

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Rozšíření MS Outlook o sofistikovaný**  
**time management**

**Time Management Outlook Plug-in**

**2010**

**Vladislav Švrčina**

# Zadání

Cílem této diplomové práce je rozšířit MS Outlook aplikaci o sofistikovaný time management systém. Rozšíření spočívá v prototypové implementaci pluginu, který se bude starat o řazení zpráv a úkolů do kategorií, řazení do kalendáře apod. Současná implementace Outlooku nesplňuje všechny požadavky, které jsou popsány pomocí metodiky "Getting things done".

1. Nastudujte metodologii "Getting things done".
2. Navrhněte její začlenění do MS Outlook.
3. Naimplementujte tuto metodu podle návrhu z bodu 2.

## Prohlášení studenta

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

V Ostravě 7. 5. 2009

.....

Plné jméno

## Poděkování

Děkuji vedoucímu bakalářské práce Ing. Svatopluku Štolfovi, Ph.D. za odborné konzultace a ochotnou pomoc při tvorbě této práce.

Také děkuji své manželce a dětem za pochopení a podporu při studiu.

## Abstrakt

Tato bakalářská práce pojednává o time managementové metodologii „Getting Things Done“, která byla vytvořena a uvedena do praxe americkým koučem a poradcem Davidem Allenem.

Rovněž se tato práce snaží aplikovat tuto metodologii do prostředí aplikace Microsoft Outlook 2007, která, přesto, že je jednou z nejvíce používaných time managementových aplikací, neposkytuje v tomto ohledu některé z potřebných funkcionalit vyžadovaných metodologií Getting Things Done.

## Klíčová slova

MS Outlook 2007, MS Office Add-In, Metodologie Getting Things Done, David Allen, Time Management, Mít vše hotovo, VSTO, Visual Studio Tools for Office

## Abstract

Current thesis is solving a time-management methodology called “Get Things Done” in MS Outlook 2007 context. This methodology was been created by American couch and consultant David Allen.

Current thesis also tries to apply this methodology into Microsoft Outlook 2007 application. Although this application is one of most used time-management applications, some possibilities are missing to fulfill the Get Things Done methodology requirements.

## Key words

MS Outlook 2007, MS Office Add-In, Getting Things Done methodology, David Allen, Time management, VSTO, Visual Studio Tools for Office

# Seznam použitých symbolů a zkratek

Symbol	Definice
COM	Common Object Model
FK	Foreign Key (Cizí klíč)
GTD	Metodologie „Getting Things Done“
MS	Microsoft
PIA	Primary Interop Assemblies
PIM	Personal Information Manager
SW	Software
VBA	Visual Basic for Applications
VSTO	Visual Studio Tool for Office

# Obsah

<b>1</b>	<b>ÚVOD .....</b>	<b>10</b>
<b>2</b>	<b>METODOLOGIE GTD .....</b>	<b>11</b>
2.1	VZNIK METODOLOGIE GTD .....	11
2.2	HLAVNÍ RYSY METODOLOGIE GTD .....	12
2.2.1	Základní cíle metodologie GTD .....	12
2.2.2	GTD a stres free produktivity .....	12
2.2.3	Horizontální a vertikální řízení činnosti .....	14
2.3	PĚT KROKŮ K OVLÁDNUTÍ PRACOVNÍHO PROCESU .....	15
2.3.1	Sesbírání .....	15
2.3.2	Zpracování .....	17
2.3.3	Uspořádání a organizace .....	19
2.3.3.1	Projekt .....	19
2.3.3.2	Zpracování dalších kroků .....	19
2.3.4	Hodnocení .....	20
2.3.5	Realizace .....	20
2.3.5.1	Model čtyř kritérií pro okamžité rozhodování o dalších krocích .....	20
2.3.5.2	Trojitý model pro hodnocení každodenní práce .....	21
2.3.5.3	Šestiúrovňový model pro hodnocení vlastní práce .....	21
<b>3</b>	<b>VÝVOJ APLIKACÍ PRO MS OFFICE .....</b>	<b>23</b>
3.1	HISTORIE A SOUČASNOST MS OUTLOOK .....	23
3.2	VÝVOJ APLIKACÍ PRO MS OFFICE .....	24
3.3	VISUAL STUDIO A VÝVOJ DOPLŇKŮ PRO MS OFFICE .....	24
3.4	ARCHITEKTURA APLIKAČNÍ VRSTVY DOPLŇKU .....	25
3.4.1	Registrace doplňku .....	26
3.4.2	Manifest rozložení a aplikační manifest .....	27
3.4.3	Visual Studio Tools for Office Runtime .....	27
3.4.4	Spolupráce doplňku a MS Office aplikací .....	28
3.5	VYTVOŘENÍ DOPLŇKU PRO MS OFFICE .....	29
3.5.1	Objektový model aplikace Outlook .....	31
3.6	ZPŮSOBY INSTALACE DOPLŇKU .....	31
3.6.1	Instalátor „ClickOne“ .....	31
3.6.2	Instalátor „Windows Installer“ .....	32

<b>4</b>	<b>IMPLEMENTACE GTD DO PROSTŘEDÍ MS OUTLOOK .....</b>	<b>33</b>
4.1	ANALÝZA IMPLEMENTACE GTD.....	33
4.1.1	<i>Analýza MS Outlooku z hlediska GTD</i> .....	33
4.1.2	<i>Specifikace požadavků pro implementaci GTD</i> .....	34
4.1.2.1	Obecné předpoklady, specifikace vstupů a výstupů .....	34
4.1.2.2	Kategorie závislé na roli uživatele.....	35
4.1.2.3	Projekty závislé na roli uživatele.....	35
4.1.2.4	Archívy.....	36
4.1.2.5	Omezení hierarchie kategorií, projektů a archívů.....	36
4.1.2.6	Outlook objekty, pravidla pro jejich zařazování.....	37
4.1.3	<i>Koncepční analýza</i> .....	38
4.1.3.1	Správa kategorií.....	38
4.1.3.2	Správa projektů .....	39
4.1.3.3	Správa archívů.....	40
4.1.4	<i>Databáze</i> .....	41
4.1.4.1	Lineární zápis.....	42
4.1.5	<i>Datová analýza</i> .....	42
4.1.6	<i>Funkční analýza</i> .....	44
4.1.6.1	Seznam elementárních funkcí.....	44
4.1.6.2	Rozdělení funkcí do subsystému (Data Flow Diagramy) .....	45
4.1.6.2.1	DFD 1. úrovně.....	45
4.1.6.2.2	DFD 2. úrovně – funkce subsystému „1. Správa“ .....	46
4.1.6.2.3	DFD 3. úrovně – funkce subsystému „1.1 role_sprava“ .....	47
4.1.6.2.3.1	Návrh algoritmu subsystému „1.1.1 role_nova“ .....	47
4.1.6.2.3.2	Návrh algoritmu subsystému „1.1.2 role_edit“ .....	48
4.1.6.2.3.3	Návrh algoritmu subsystému „1.1.3 role_odstran“ .....	48
4.1.6.2.4	DFD 3. úrovně – funkce subsystému „1.2 kat_sprava“ .....	49
4.1.6.2.4.1	Návrh algoritmu subsystému „1.2.1 kat_nova“ .....	49
4.1.6.2.4.2	Návrh algoritmu subsystému „1.2.2 kat_edit“ .....	51
4.1.6.2.4.3	Návrh algoritmu subsystému „1.2.3 kat_odstran“ .....	51
4.1.6.2.5	DFD 3. úrovně – funkce subsystému „1.3 proj_sprava“ .....	52
4.1.6.2.5.1	Návrh algoritmu subsystému „1.3.1 proj_novy“ .....	52
4.1.6.2.5.2	Návrh algoritmu subsystému „1.3.2 proj_edit“ .....	54
4.1.6.2.5.3	Návrh algoritmu subsystému „1.3.3 proj_odstran“ .....	55
4.1.6.2.5.4	DFD 3. úrovně – funkce subsystému „1.4 arch_sprava“ .....	56
4.1.6.2.5.5	Návrh algoritmu subsystému „1.4.1 arch_novy“ .....	56
4.1.6.2.5.6	Návrh algoritmu subsystému „1.4.1 arch_edit“ .....	58
4.1.6.2.5.7	Návrh algoritmu subsystému „1.4.2 arch_odstran“ .....	58
4.1.6.2.6	DFD 3. úrovně – funkce subsystému „1.5 obj_sprava“ .....	59



4.1.6.2.6.1	Návrh algoritmu subsystému „1.5.1 obj_kategorie“	59
4.1.6.2.6.2	Návrh algoritmu subsystému „1.5.2 obj_projekt“	64
4.1.6.2.6.3	Návrh algoritmu subsystému „1.5.3 obj_archív“	66
4.1.6.2.7	DFD 2. úrovně – funkce subsystému „2 report“	69
4.1.6.2.7.1	Návrh algoritmu subsystému „2.1 rep_kategorie“	69
4.1.6.2.7.2	Návrh algoritmu subsystému „2.2 rep_projekt“	70
4.1.6.2.7.3	Návrh algoritmu subsystému „2.1 rep_archive“	72
4.1.6.3	Diagramy aktivit	74
4.1.6.4	Diagram aktivit subsystému „1.5 objekt_sprava“	74
4.1.6.5	Diagram aktivit subsystému „2 report“	75
4.1.7	Analýza datových toků	76
4.1.8	Indexová analýza	76
4.1.9	Transakční analýza	76
4.2	NÁVRH IMPLEMENTACE DOPLŇKU GTD	77
4.2.1	SW prostředky pro implementaci doplňku	77
4.2.2	Návrh úložišť a konfiguračních souborů doplňku	77
4.2.3	Návrh grafického rozhraní doplňku	79
4.2.3.1	GTD v menu aplikace MS Outlook 2007	79
4.2.3.2	Přehledový panel GTD	79
4.2.3.3	Nová lišta nástrojů (Ribbon) GTD	80
4.2.3.4	Dialogy pro správu kategorií, projektů a archívů	81
4.2.3.5	Dialogy pro úpravu rolí, kategorií, projektů a archívů	83
4.2.4	Návrh implementace rozhraní a tříd	84
4.3	IMPLEMENTACE DOPLŇKU GTD	86
4.3.1	Skladba použitých tříd a rozhraní	86
4.3.2	Jmenné prostory	87
4.3.3	Konfigurační soubor	88
4.3.4	Získávání EntryID	88
4.4	ŘEŠENÍ PROBLÉMU PŘI IMPLEMENTACI GTD	89
4.4.1	MS Outlook Security Issue	89
4.4.2	MS Visual Studio a SPI	89
<b>5</b>	<b>ZÁVĚR</b>	<b>90</b>
	<b>SEZNAM POUŽITÉ LITERATURY</b>	<b>92</b>
	<b>SEZNAM PŘÍLOH</b>	<b>93</b>

# 1 Úvod

V dnešní hektické době je většina lidí doslova zahlcena různými povinnostmi a úkoly, které se jim v mnoha případech, i přes jejich veškerou snahu, nedaří zorganizovat tak, aby mohli svým závazkům dostát. Problémy pramenící z nedostání závazkům jsou pak stress, časová tíseň, nesoustředěnost a mohli bychom pokračovat dál ve výčtu negativních dopadů. Všechny tyto aspekty pak mají přímý nebo nepřímý vliv na náš pracovní i rodinný život, na naši duševní pohodu, a pokud jsou stresové situace vyplývající z výše uvedeného permanentní, je zde i velká pravděpodobnost negativního vlivu na naše zdraví.

Aktuální trh obsahuje mnoho prostředků, jak hardwarových tak softwarových, které mají za cíl pomoci nám efektivně zorganizovat náš čas. Řada těchto produktů již prošla několikaletým vývojem a v dnešní době již disponují sofistikovanými metodami nejmodernějšího time managementu.

Tato práce se bude zabývat výhradně softwarovými prostředky k zvýšení výkonnosti a organizaci času, které nabízí aktuální trh. Konkrétněji se pak práce bude věnovat produktu Microsoft Outlook, firmy Microsoft corporation, který je nejvíce používaným emailovým a groupwarovým klientem vůbec.

Cílem práce je:

- seznámení se s metodologií „Getting Things Done“;
- analýza aplikace MS Outlook 2007 z pohledu metodologie GTD;
- návrh implementace metodologie GTD do aplikace MS Outlook 2007;
- nalezení způsobu, jak navrženou implementaci GTD zavést do prostředí aplikace MS Outlook 2007;
- vytvoření implementace metodologie GTD a její zavedení do aplikace MS Outlook 2007;

## 2 Metodologie GTD

Cílem této kapitoly není vyčerpávající popis metodologie „Getting Things Done“ Davida Allena, který můžete v detailní podobě nalézt ve stejnojmenné knize[1], ale pouze nastínění základních rysů a zásad, jenž z ní činí nejvíce efektivní time managementovou metodu současnosti.

### 2.1 Vznik metodologie GTD

Motto: „Univerzální prostředek, jak dovést osobní produktivitu a organizaci k dokonalosti, sice neexistuje, existují nicméně postupy, které vám to usnadní.“ [1]

Více než dvacet let se Američan David Allen, jenž aktuálně působí jako manažerský poradce, snažil vyvinout efektivní systém pro organizaci pracovních záležitostí. Bádal nad vývojem nových metod v oblasti prostředků rozvoje osobní produktivity a organizace, a toto bádání jej dovedlo k přesvědčení, že žádný univerzální nástroj nebo řešení, které by vždy a za každých okolností fungovalo, neexistuje. Toto je do určité míry způsobeno měnícím se pracovním prostředím a novými výzvami, kdy zjišťujeme, že námi zažité metody nám přestávají poskytovat potřebnou jistotu, že jimi dosáhneme cílů, které jsme si vytyčili.

Z mnoha tisíců hodin, které strávil David Allen koučováním lidí přímo v terénu, kdy jim pomáhal zvládat organizaci každodenních povinností, vzešly postupy, které jak autor uvádí „se osvědčily jako vysoce efektivní ve všech typech organizací, na každé úrovni firemní hierarchie, napříč kulturami, dokonce i v domácnostech a ve školách.“ [1].

Dnešní pracovní tempo způsobuje, že mnoho věcí se nám během standardní pracovní doby nepodaří plně vyřešit. Mnoho záležitostí jako došlá pošta, telefonní hovory nebo emaily zůstává nevyřízeno. Tyto záležitosti nás pak nutí o nich neustále přemítat ve své mysli, a to bohužel i v čase, kdy již nemáme možnost tyto záležitosti nějakým způsobem řešit. Proto nám David Alen nabízí metodologii „Getting Things Done“ (dále již GTD) a jejich několik zásad k tomu, abychom dokázali svůj čas a energii lépe využít.

## **2.2 Hlavní rysy metodologie GTD**

Člověk si může udržet produktivitu, i pokud je pracovně velmi vytížen. Vše ale vyžaduje jasnou organizaci, držení se pevných zásad a především znát a mít nastavené své priority.

Většina lidí má povědomí o tom, co je třeba podniknout, aby se daná záležitost posunula vpřed, co je potřeba k úspěšnému zakončení v tom či onom projektu apod. Tato většina lidí, ale mnohdy potřebuje to, co ví, také převést do každodenní reálné praxe. Dokázat uplatňovat nabyté zkušenosti s lepším načasováním a komplexně. I drobné změny v organizaci naší práce mohou znamenat velký posun vpřed.

### **2.2.1 Základní cíle metodologie GTD**

V tomto ohledu má metodologie GTD dva základní cíle:

- všechny záležitosti, které máme zvládnout (aktuálně nezáleží kdy přesně - jestli teď, nebo později, ani nezáleží na důležitosti daného úkolu) musíme mít zaznamenané v systému, kterému plně důvěřujeme. Nic nesmí zůstat v naší hlavě – jediné tak se zbavíme neustálých myšlenek na to, co máme dělat, co jsme zapomněli apod. – víme, že to co máme udělat, je zaznamenáno v důvěryhodném systému a kdykoli zde nahlédneme, budeme přesně vědět, co v této chvíli je nejdůležitější;
- dodržovat sebekázeň, abychom pro všechna rozhodnutí, která jsme přijali k řešení, měli plán dalšího postupu a rovněž dokázali tento plán účelně spravovat. To znamená, abychom dokázali tento plán v případě potřeby upravit a přizpůsobit novým, a stále měnícím se, požadavkům;

### **2.2.2 GTD a stres free produktivity**

Stres, který je přítomný v našem životě, pak velkou mírou pochází právě z nevyřešených záležitostí, které nám byly přiděleny nebo jež jsme přijali. Pokud zvládneme efektivně řešit tyto otevřené záležitosti, dostane náš život nový produktivnější ráz. Budeme se cítit více uvolnění a i schopnost koncentrace a soustředěnosti se výraznělepší.

Abychom se dokázali při přejímání závazků chovat efektivně, je nutné vše, co je nějakým způsobem svázáno s naší prací a vyžaduje naši pozornost, dokázali co nejvhodněji klasifikovat, zatřídit a samozřejmě naplánovat další kroky v dané záležitosti. V praxi však

nejsme mnohdy v tomto ohledu vyhranění a jednoznační. Mnohdy váháme a není jednoduché danou věc správně klasifikovat.

K tomu abychom v těchto případech postupovali opravdu efektivně, je zapotřebí si uvědomit následující:

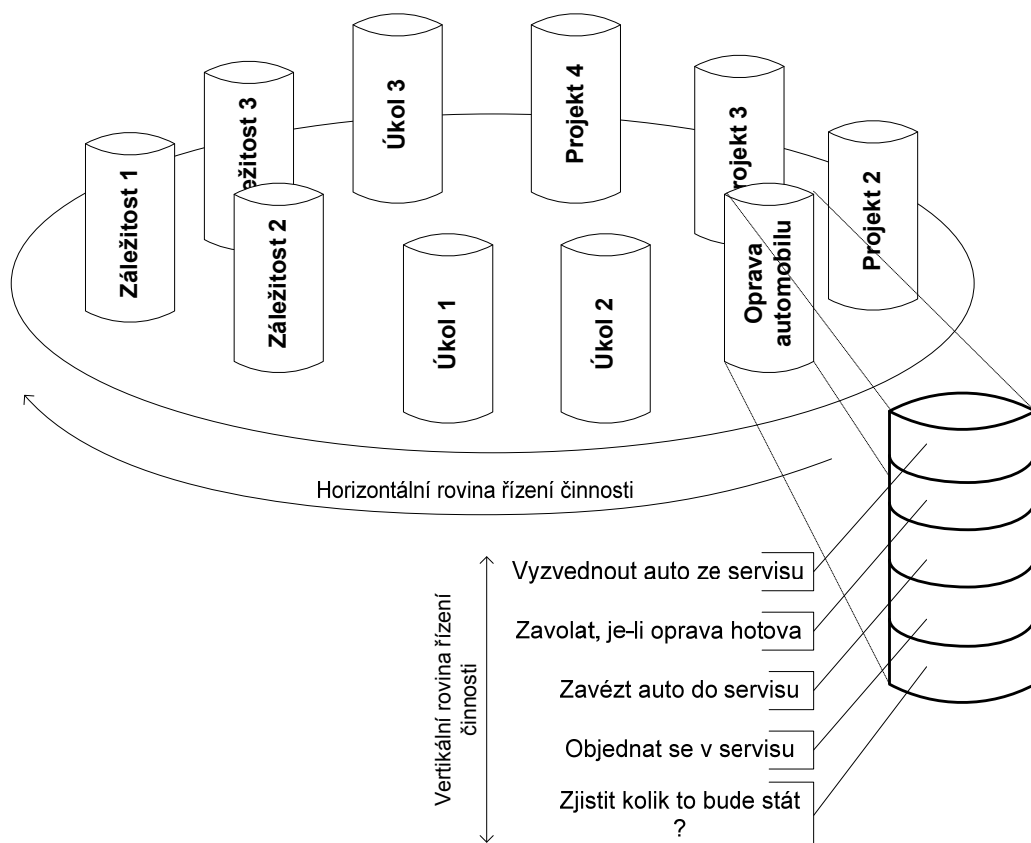
- Vše co je zapotřebí udělat nebo dokončit musí být zaznamenáno v důvěryhodném systému (autor [1] hojně využívá pojmu „v systému mimo hlavu“) jako např. sběrná přihrádka nebo schránka, kterou pravidelně vybíráme
- Uvědomit si o čem je daný závazek, jaké povinnosti a následující další kroky z něj vyplývají, tedy co je potřeba s daným závazkem udělat
- Jakmile si uvědomíme další nezbytné kroky, které z přijatého závazku vyplynuly, je nezbytné také tyto kroky zaznamenat do systému, který pravidelně sledujeme

Při přejímání závazku a definici dalších kroků bychom si měli vždy uvědomit co je ke zvládnutí dané záležitosti potřebné, jaký „konkrétní“ další krok je nutný k tomu, abychom se v dané otázce posunuli vpřed, popř. ji vyřešili. Nevytvářejme nikdy obecné kroky, které nám nic nevypoví o důležitosti závazku nebo nám neposkytnou jasnou instrukci co je zapotřebí vykonat. Uveďme si příklad z reálné praxe – pracujeme na projektové dokumentaci stavby domu – máme rozpracovanou část elektroinstalace – zadavatel nás kontaktuje prostřednictvím emailu a požaduje, abychom počítali s připojením dalšího spotřebiče, který má vyšší odběr proudu. Jaké by měly být v tomto případě další kroky? Určitě nejprve zjistíme aktuální odběr všech běžně používaných spotřebičů v domácnosti a spotřebičů, které zadavatel plánuje do domácnosti instalovat, a porovnáme s instalovaným příkonem domu. Dalším krokem by mělo být kontaktovat energetické závody v případě nutnosti navýšení příkonu a získat informaci, zdali je možno v požadované lokalitě daný příkon získat a o tento pak požádat. Tyto kroky nám ukazují konkrétní úkoly, které je nutno udělat k vyřešení této záležitosti.

Zjednodušeně řečeno, základem úspěchu je přemýšlet o práci dříve, než se do ní pustíme a ujasnit si co pro danou práci potřebujeme. Následně tyto další kroky přesně specifikovat a zapsat do již zmíněného systému - takzvaně je vypustit z hlavy.

Musíme pamatovat, že naši mysl nelze ošálit a pokud neprovedeme vše co bylo zapotřebí (i když si ujasníme další kroky, ale neprovedeme zápis do systému), budeme stále pociťovat nutkání o dané záležitosti přemýšlet znovu a znovu, čímž nejen zbytečně ztrácíme čas, ale rovněž mrháme svou energií opakovaným přemítáním o dané záležitosti a prohlubujeme svůj stres a frustraci.

### 2.2.3 Horizontální a vertikální řízení činnosti



Obr. 1: Příklad horizontálního a vertikálního řízení činnosti

Naše závazky musíme sledovat ve dvou rovinách – horizontální a vertikální.

Horizontální rovinou jsou myšleny všechny záležitosti, projekty, úkoly okolo nás, v nichž se angažujeme.

Za vertikální rovinu je pak považován detailní pohled na konkrétní záležitost nacházející se v naší horizontální rovině.

Když si osobně představíme naše záležitosti v těchto dvou rovinách, dokážeme získat poměrně komplexní přehled o našich závazcích. Z pohledu těchto dvou rovin se nám bude zdát převedení všech nevyřešených záležitostí do systému jednodušší a jejich konkrétní specifikace jasnější. Zmíněné rozdělení do horizontální a vertikální roviny nám opětovně pomáhá přenést veškeré naše závazky do systému mimo naši mysl. To je také hlavním cílem GTD – provést klasifikaci a uložení 100% našich záležitostí do pomocných systémů mimo naši hlavu.

## 2.3 Pět kroků k ovládnutí pracovního procesu

Metodologie GTD specifikuje pět kroků, jimiž každý pracovní proces prochází:

1. **Sesbírání** věcí...
2. **Zpracování** jejich významu a určení dalších kroků...
3. **Uspořádání** výsledků a...
4. **Hodnocení**, jejich variant, zdali jdou v daném okamžiku realizovat...
5. a poté vlastní **Realizace**

Tyto jednotlivé kroky je vhodné od sebe navzájem oddělovat, tj. neprovádět všechny najednou. Obvykle nebývá z časového hlediska proveditelné provést právě všech pět kroků v daném okamžiku a v tomto případě by naše úsilí mohlo přijít vniveč.

Vhodnější je nejdříve sesbírat informace bez jakéhokoli třídění. Zpracování, třídění, hodnocení atd. lze provádět postupně po jednotlivých částech, jak nám to aktuální pracovní vytíženost dovolí. Samozřejmě, že bychom měli postupně zpracovávat všechny vstupy, nikoli jen ty co považujeme za nejdůležitější, protože ty „méně“ důležité nezpracované vstupy nám pak zůstávají v mysli a brání efektivitě naší práce.

### 2.3.1 Sesbírání

Jak již bylo zmíněno v předchozích kapitolách, prvním a nejdůležitějším krokem, jak vytěsnit nedokončené záležitosti z naší mysli je tyto informace zanešt do externího systému mimo naši hlavu. Abychom to mohli provést, musíme veškeré rozpracované a nedokončené záležitosti sesbírat a shromáždit v tzv. vstupní schránce, která dokáže zachytit jak naše vstupy, tak vstupy přicházející z externích zdrojů.

Pod pojmem vstupní schránky si lze představit např. plastovou přihrádku na došlou poštu, jako dopisy, časopisy a jiné hmotné věci. Používáme-li emailového klienta, pak vstupní schránkou může být inbox složka, čili složka s doručenou poštou. Jako schránku můžeme také používat obyčejnou složku na papíry, na které si budeme psát poznámky, nápady atp.

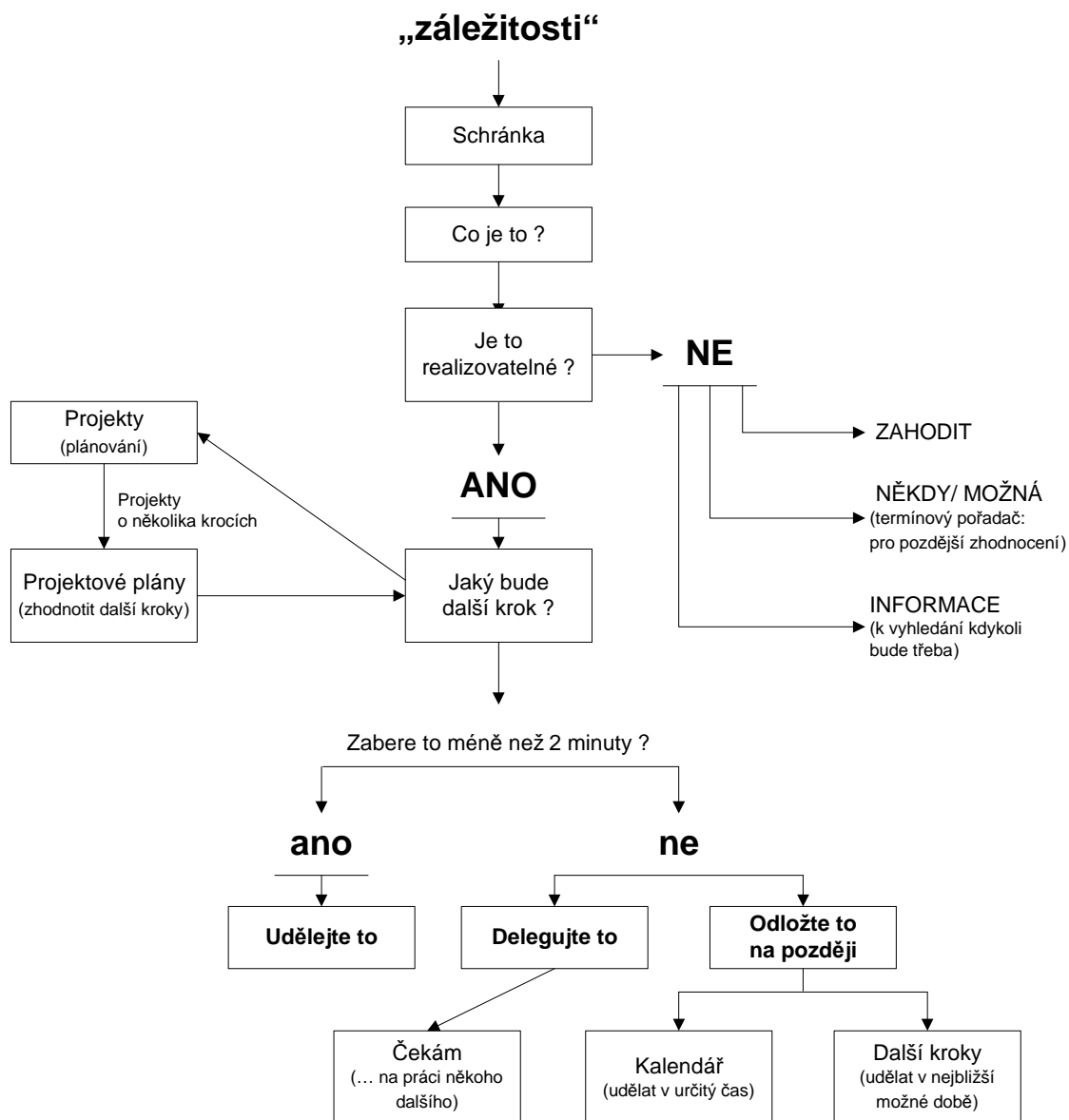
Těchto schránek lze mít více než jednu. Obecně však platí pravidlo, že čím méně tím lépe. Každopádně nutnou podmínkou funkčních vstupních schránek je jejich pravidelné vyprazdňování.

Vyprázdněním vstupní schránky zde není myšleno dokončení záležitostí, které jsou ve schránce obsaženy, ale jejich zpracování a zařazení dále do systému. Co ze schránky vyzvedneme, již nikdy nevracíme zpět! Takové jednání by bylo kontraproduktivní – pokud např. přečteme doručený email, pak správný postup je jeho zhodnocení a zařazení do dalšího procesu, nikoli jeho ponechání v příchozí poště. Ponecháme-li email v složce pro příchozí poštu, příště ztratíme čas s jeho opětovným čtením, ohodnocením a zařazením.



### 2.3.2 Zpracování

V této fázi procesu hodnotíme záležitosti vyzvednuté ze vstupní schránky. Vlastní systém hodnocení je prezentován následujícím modelem rozhodovacího stromu (Obr. 2: Model rozhodovacího stromu GTD [1]):



Obr. 2: Model rozhodovacího stromu GTD [1]

U každé záležitosti, kterou se snažíme zpracovat, je důležité si uvědomit, o co v téhle záležitosti jde (položka „Co to je?“). Když víme co je jádrem této záležitosti, můžeme odpovědět na jednoduchou otázku: „Je záležitost realizovatelná?“

V případě, že si odpovíme „Ne“ můžeme se rozhodnout pro jednu ze tří možností, jak danou záležitost vyřešíme:

1. záležitost už nebude potřeba, můžeme to vyhodit
2. v budoucnu se mohou informace obsažené v této záležitosti hodit, záležitost tedy archivujeme
3. záležitost není nyní aktuální, ale v budoucnu je možnost záležitost znovu otevřít a přehodnotit, je vhodné záležitost zaznamenat do diáře nebo termínového pořadače pro připomenutí

Pokud je ovšem naše odpověď na otázku realizovatelnosti zpracovávané záležitosti „Ano“, pak zvážíme, náleží-li daná záležitost k nějakému již existujícímu projektu, popř. jestli zpracovávaná záležitost vyžaduje vytvoření projektu nového.

Rozhodovací model GTD dále pokračuje procesem „Jaký bude další krok?“. Krokem, jak jsme si již vysvětlili dříve, je další konkrétně viditelná fyzická činnost, kterou zajistíme posun v realizaci dané záležitosti. Důležité je také zvážit, zdali zpracovávaná záležitost vyžaduje k realizaci pouze jeden jediný krok, nebo bude zapotřebí více dalších kroků.

Jakmile bylo rozhodnuto o dalším kroku, nabízí se nám další tři možnosti, kudy se ubírat:

1. Udělejte to – zde jde o to, pokusit se danou záležitostí vyřešit okamžitě, zabere-li nám její řešení méně než dvě minuty. Obecně můžeme říci, pokud takovou záležitost vyřídíme okamžitě, zabere nám to méně času, než bychom vynaložili na další zpracování;
2. Delegujte to – zvažme, zdali adekvátní akce bude trvat déle, než ony dvě minuty a uvědomme si zde, zdali jsme tou správnou osobou, které se řešení této záležitosti týká. Je-li naše odpověď záporná, pak předejme tuto záležitost kompetentní osobě. Delegujme ji;
3. Odložte to na později – zabere-li nám realizace této záležitosti více než dvě minuty a jsme opravdu osobou odpovědnou za řešení, pak danou záležitost odložíme na později. Samozřejmě je důležité vytvořit seznam dalších kroků v našem systému

### **2.3.3 Uspořádání a organizace**

V předchozí kapitole jsme vyloučili záležitosti, které nepotřebují žádnou akci. Tyto záležitosti byly buďto vymazány, archivovány nebo uloženy jako připomínka v případě, že v budoucnu zvažujeme danou záležitost znovu otevřít.

Tímto filtrem by nám tedy měly projít jen záležitosti, vyžadující další akci.

Abychom úspěšně mohli administrovat tyto akční záležitosti, je nutné mít seznam projektů, seznam dalších kroků a seznam věcí, na které čekáme. Pro věci týkající se projektů, jako jsou různé nákresy a plány, je vhodné mít pro každý projekt zvlášť složku, kde budeme tyto související věci ukládat. Důležité opět je, aby vše co zpracujeme, bylo uloženo v důvěryhodném systému mimo naši hlavu.

#### **2.3.3.1 Projekt**

Pokud si vyřešení záležitosti vyžádá více než jeden konkrétní krok, je vhodné založit k dané problematice projekt. Vytvořený projekt pak shromažďuje konkrétní kroky, které k jeho úspěšnému splnění musíme zrealizovat.

I když některé záležitosti nevypadají, že by bylo nutné vytvářet projekt k jejich dokončení, v praxi se osvědčilo, tyto projekty založit. Pokud totiž danou záležitost opravdu nerozdělíme do dalších kroků, budeme muset řešení problému nosit opět v hlavě, a to je proti zásadám GTD.

Samotný projekt nelze přímo realizovat. Realizujeme jen aktuální dílčí kroky a tím se posouváme v realizaci projektu dále. Jakmile dokončíme definované kroky projektu a aktuální projekt se shoduje s našimi představami, které jsme o projektu měli při jeho zakládání, pak můžeme projekt prohlásit za hotový.

#### **2.3.3.2 Zpracování dalších kroků**

Většina záležitostí jde rozdělit do dvou kategorií.

Jedna kategorie představuje záležitosti, které musíme vyřídit v určitém dni nebo určitém čase. Pro tyto záležitosti je ideální použití diáře. Metodologie GTD ovšem varuje před vkládáním seznamu úkolů na daný den do diáře. Tato běžná praxe tradičních time managementu zde nefunguje, a to hlavně z důvodů hektického pracovního tempa současnosti. U mnoha úkolů, které chceme zvládnout během dne, jednoduše nemáme možnost je dokončit. Většinou je tato situace zapříčiněna neustálými změnami v prioritách, ke kterým během dne dochází. Z tohoto

důvodu je pak následné přesouvání nesplněných úkolů na další dny zbytečnou ztrátou času. Navíc nesplněné úkoly, které neustále odkládáme, nás vnitřně demoralizují.

Do druhé kategorie spadají záležitosti, které nemají přesně stanovený čas ani datum, ale je zapotřebí se jim věnovat co nejdříve. Tyto položky patří na zvláštní seznam dalších kroků. Obsahuje-li tento seznam mnoho položek (více než cca 30), pak je vhodná další kategorizace dalších kroků z důvodu lepší orientace (např. „Telefonáty“, „Doma“, „Na poradě“ apod.).

### **2.3.4 Hodnocení**

V určitých intervalech se musíme vracet zpět a sebekriticky a s nadhledem zhodnotit naše životní a pracovní záležitosti.

Doporučené je tzv. „týdenní“ hodnocení, kdy na konci běžného pracovního týdne projdeme všechny nedokončené projekty a otevřené záležitosti. Zde se nám naskýtá možnost podívat se na dosavadní průběh realizace, zvážit jestli se ubírá správným směrem a případně přehodnotit další kroky.

Toto hodnocení je rovněž důležité k aktualizaci údajů v našem systému, protože ve většině případů nalezneme při procházení otevřených záležitostí ty, které potřebují některé informace upřesnit. Druhý aspekt této aktualizace je, že udržujeme náš systém „živý“ a „úplný“ a tedy víme, že na informace v něm obsažené se můžeme 100% spolehnout.

### **2.3.5 Realizace**

Není vždy lehké vybrat z aktuálního seznamu dalších kroků, ten, který budeme právě teď realizovat. Můžeme se řídit svou intuicí nebo nám mohou pomoci tři modely pro rozhodování, co budeme dělat:

#### **2.3.5.1 Model čtyř kritérií pro okamžité rozhodování o dalších krocích**

Měli bychom zachovat posloupnost těchto čtyř kritérií:

1. *Kontext* – ne všechny typy činnosti můžeme dělat kdekoli. Většinou jsme vázáni na konkrétní místo nebo pracovní prostředek

2. *Dostupný čas* – musíme zvážit, jaký časový prostor máme aktuálně k dispozici. Je neefektivní začít déle trvající práci, kterou budeme muset za 15 minut přerušit, např. kvůli meetingu
3. *Dostupná energie* - energie je důležitá u práce, která je fyzicky či duševně pro nás namáhavá, vyžaduje soustředěnost apod., v protipólu jsou pak práce, které nemají velké nároky na naši energii
4. *Priorita* – zde se musíte víceméně spolehnout na vaši intuici, zvážit, do které práce je v tento okamžik nejlepší investovat svou energii. Zbývající dva modely nám pak usnadní toto rozhodování

#### **2.3.5.2 Trojitý model pro hodnocení každodenní práce**

Své závazky realizujeme podle těchto typů aktivit:

- *Děláme předem určenou práci* – v tomto případě se zabýváme prací, která přímo vyplývá z aktuálního seznamu dalších kroků, který jsme si vytvořili
- *Děláme práci, která se objevila* – některé záležitosti se objevují náhodně a nepředvídatelně, což je pro chod světa, jak jej známe, charakteristické. Je obvyklé, že vznikne situace, kdy je zapotřebí řešit věci, které jednoduše chybí na našem seznamu dalších kroků. Zde musíme opět zapojit svou intuici a zvážit, zdali je nově se objevivší věc takové důležitosti, že se ji musíme okamžitě začít věnovat, nebo ji stačí aktuálně pouze zpracovat do našeho seznamu dalších kroků a můžeme se vrátit k původní práci
- *Definujeme svou práci* – zde může být zahrnuto např. zpracování vstupní schránky, aktualizace a hodnocení v rámci našeho systému, apod.

#### **2.3.5.3 Šestiúrovňový model pro hodnocení vlastní práce**

Tento model nabízí z pohledu analogie z letectví, kvalitnější rozlišení priorit, než poskytují současné time managementové metodologie:

- *Nad 50 000 stop: Život* – představuje vnímání našeho jednání a našich cílů s velkým nadhledem s nádechem filosofičnosti. Určujeme si celkové pracovní a rodinné cíle, od kterých se pak odvíjí cíle našich nižších „letových“ hladin

- *40 000 stop: Vize na 3-5 let* – projekty z dlouhodobější perspektivy. Zde určujeme naše životní priority a názory na práci, rodinu, život a k těmto cílům se snažíme naše jednání směřovat. Naše plány v této úrovni mívají zásadní vliv na ostatní nižší úrovně
- *30 000 stop: Cíle na příští 1 - 2 roky* – projekty, které se týkají cílů, jichž chceme dosáhnout v průběhu 1 - 2 let. Většina naší práce a to, jakým způsobem ji realizujeme, nás pak směřuje k vytyčeným cílům;
- *20 000 stop: Oblasti zodpovědnosti* – jedná se většinou o projekty, které jsme přejali z hlediska naší osobní zodpovědnosti. Mohou se dotýkat nejen pracovního, ale i osobního života. Jsou zde projekty z hlediska osobního růstu a rozvoje, financí, rodinného života apod.
- *10 000 stop: Aktuální projekty* – relativně krátkodobé záležitosti (projekty) o více krocích, které máme plánu zvládnout v brzké době
- *Vzletová dráha: Aktuální aktivity* – představuje seznam věcí, které aktuálně čekají na naši reakci a jež bychom měli zvládnout

## 3 Vývoj aplikací pro MS Office

Tato část práce se nevěnuje přímo popisu sady MS Office jako takové, ale primárně se zaměřuje pouze na produkt MS Outlook. Nicméně hlavním cílem této kapitoly je seznámit nás s počátky i současnými trendy ve vývoji aplikací pro MS Office.

### 3.1 Historie a současnost MS Outlook

Historie samotného produktu Microsoft Outlook se datuje od roku 1997, kdy byla aplikace ve verzi Outlook 97 oficiálně uvedena na trh jako součást kancelářského balíku Microsoft Office 97. Předchůdcem MS Outlooku byly aplikace Schedule+ a Exchange Client. Aplikace Schedule+ byla poprvé uvedena s operačním systémem Windows 3.11 v roce 1992. Exchange Client byl ještě použit jako součást Exchange Serveru 4.0 a 5.0 z roku 1996 a 1997. Exchange Server 5.5 již neobsahoval aplikaci Exchange Client a oficiální náhradou za ni se stal právě MS Outlook.





Již ve verzi 97 Outlook obsahoval emailového klienta, kontakty, kalendář s úkoly, a poznámky. Tím se stal tzv. plnohodnotnou PIM aplikací.

Od doby prvního nasazení MS Outlook až po dnešní aktuálně dostupnou verzi 2007, odvedla firma Microsoft pořádný kus práce.

Těžkopádné ovládání původního Office 97 prošlo řadou změn, ať se to týká funkčnosti, bezpečnosti nebo úprav grafického rozhraní aplikace. Firma Microsoft vždy klade do popředí ergonomičnost ovládání svých aplikací. Její dlouholeté zkušenosti se zúročily hlavně v poslední sadě kancelářských aplikací MS Office 2007, které disponují kompletně přepracovaným systémem ovládání. Prvotní kritika nového GUI vycházela spíše z radikálnosti změn, než z jemu vytýkané nevhodné implementace.

Oficiálně je v době psaní této práce jako poslední dostupná verze 2007. Nicméně verze 2010 je již dokončena a její uvolnění je plánováno v následujícím měsíci, tedy květnu roku 2010.

**Tabulka 1: Přehled aktuálně vydaných verzí MS Outlook [2]**

Ikona	Oficiální název	Majoritní číslo verze	Datum uvolnění
	Outlook 97	8.0	Leden 1997
	Outlook 98	8.5	Červen 1998
	Outlook 2000	9.0	Červen 1999
	Outlook 2002	10.0	Květen 2001
	Outlook 2003	11.0	Říjen 2003
	Outlook 2007	12.0	Listopad 2006

### 3.2 Vývoj aplikací pro MS Office

Programátoři mohou již pár let vyvíjet aplikace pro MS Office, ale až s příchodem MS Office 2003 se objevila skutečně prvotřídní vývojová platforma.

Už dříve bylo možno rozšířit funkčnost produktů sady MS Office o vlastní aplikace využívající záznam makra v kombinaci s MS Visual Basic for Application (VBA), popřípadě šlo vytvářet doplňky (add-in) založené na COM s použitím MS Visual Studia 6.0 nebo MS Visual C++. Až MS .NET Framework s MS Office Primary Interop knihovnami (PIA) umožnil vytvářet doplňky a smart client řešení využívajících řízený kód (managed code) objektového modelu Office aplikací založeného na COM.

### 3.3 Visual Studio a vývoj doplňků pro MS Office

Visual Studio umožňuje vytvářet a přizpůsobovat aplikace sady MS Office aktuálním požadavkům uživatele. Obecně rozlišujeme dva typy vývoje aplikací pro Office:

1. Vývoj na bázi dokumentů – tento typ vývoje je přímo zaměřen na práci s konkrétním typem dokumentu (.xls, .doc, .ppt, atd) – uživatelem vytvořený programový kód je načten v případě otevření asociovaného dokumentu. Funkce, které jsou součástí takto vytvořeného programu, jsou přístupné, pouze pokud je daný dokument otevřen. Tento způsob vývoje má jistá omezení (např. nelze



provádět změny v samotné aplikaci, tj. nelze modifikovat menu aplikace, nástrojové lišty apod.)

2. Vývoj na bázi aplikací – tento typ vývoje aplikací, je zaměřen primárně na samotné rozšíření funkčnosti aplikací sady MS Office (Word, Excel, Power Point, atd.). Zdrojový kód takto vytvořeného programu (doplňku – Add-in) je asociován přímo s danou aplikací MS Office. Většinou se doplněk spouští automaticky při startu asociované aplikace. Funkce, které doplněk aplikuje, jsou pak přístupné v této aplikaci, nezávislé na právě otevřeném dokumentu. Řešení založené na bázi aplikace umožňuje zasahovat do prostředí samotné aplikace a přizpůsobovat jej vlastním potřebám. Zároveň u tohoto typu vývoje je podporováno i použití interního nástroje pro ladění (tzv. debugger) prostředí MS Visual Studia.

Kromě těchto výše zmíněných postupů jde vytvářet ještě vlastní aplikace (jak bylo zmíněno v předchozí kapitole), které využívají tzv. Primary Interop Assemblies (PIA), což jsou knihovny, instalované jako součást MS Office. Tyto knihovny jsou uloženy v Global Assembly Cache (GAC) systému a umožňují spolupráci (interakci) s aplikacemi MS Office.

Pro vývoj doplňku pro MS Outlook 2007 je nejvhodnějším řešením právě řešení ad. 2

### **3.4 Architektura aplikační vrstvy doplňku**

Vytváříme-li doplněk pro MS Office pomocí VSTO, pak vlastně vytváříme knihovnu, obsahující náš kód, která je automaticky načtena při spuštění dané Office aplikace. Jakmile je tato knihovna načtena, dokáže zpracovávat události aplikace, které jsou v doplňku aplikovány, dokáže změnit prostředí aplikace, automatizovat některé funkce původní aplikace apod. Záleží co je v doplňku aplikováno.

Vlastní komunikace mezi doplňkem a COM komponenty aplikace pak probíhá skrz PIA aplikace.

Má-li daná aplikace instalováno více doplňků, je každý doplněk spouštěn ve vlastní aplikační doméně. Tím je zajištěna stabilita hlavní aplikace a také ostatních doplňků. V případě nekorektního chování některého z doplňků, dojde pouze k pádu daného doplňku, a ostatní doplňky včetně hlavní aplikace nejsou tímto dotčeny.

Dokumentace MSDN[4] také zmiňuje omezení pro využití doplňku, kdy doplňky vytvořené pomocí VSTO jsou určeny pro použití s hostitelskou MS Office aplikací, která je

spouštěna přímo koncovým uživatelem. Pokud je aplikace spuštěná nějakým způsobem automaticky, pak nemusí doplněk pracovat správně.

### 3.4.1 Registrace doplňku

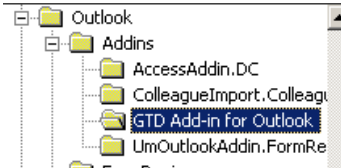
Aby mohla aplikace MS Office najít asociované doplňky, pak tyto doplňky musí být registrovány, tzn., musí být informace o těchto doplňcích zaneseny ve vstupech registrů. Zde je nutno si uvědomit, zdali bude doplněk použitelný pouze pro uživatele, který doplněk instaloval (HKEY\_CURRENT\_USER), nebo pro všechny uživatele (HKEY\_LOCAL\_MACHINE). Aplikace MS Office 2007 defaultně vyhledávají pouze doplňky registrované pod HKEY\_CURRENT\_USER. Aby aplikace aktivně vyhledávala i doplňky v klíči HKEY\_LOCAL\_MACHINE musí být na počítači, kde je provozována aplikace instalován hotfix s číslem 976477.

Konkrétní informace o doplňku jsou pro aplikaci MS Outlook 2007 uloženy v proměnných registru:

HKEY\_CURRENT\_USER\Software\Microsoft\Office\Outlook\Addins\název doplňku

Důležité informace jsou pak uloženy v následujících klících výše zmíněného registru:

- *Description* – popis doplňku
- *FriendlyName* – uživatelsky přívětivý název doplňku
- *LoadBehavior* – tato hodnota udává, kdy má aplikace načíst daný doplněk a současný stav doplňku (je-li nahrán do aplikace nebo ne). Defaultní hodnota =3 znamená, že doplněk bude načten do aplikace při startu. Pokud z nějakého důvodu aplikaci nepovede korektně zavést požadovaný doplněk, tato hodnota se automaticky změní (většinou na 2)
- *Manifest* – plná cesta k instalovanému doplňku (jeho umístění na disku, v síti apod.)



Název	Typ	Data
(Výchozí)	REG_SZ	(Hodnota není zadána)
Description	REG_SZ	GTD Add-in for Outlook - Outlook add-in ...
FriendlyName	REG_SZ	GTD Add-in for Outlook
LoadBehavior	REG_DWORD	0x00000003 (3)
Manifest	REG_SZ	C:\Documents and Settings\Test\Dokum...

Obr. 3: Registrace doplňku v registrech systému Windows

### 3.4.2 Manifest rozložení a aplikační manifest

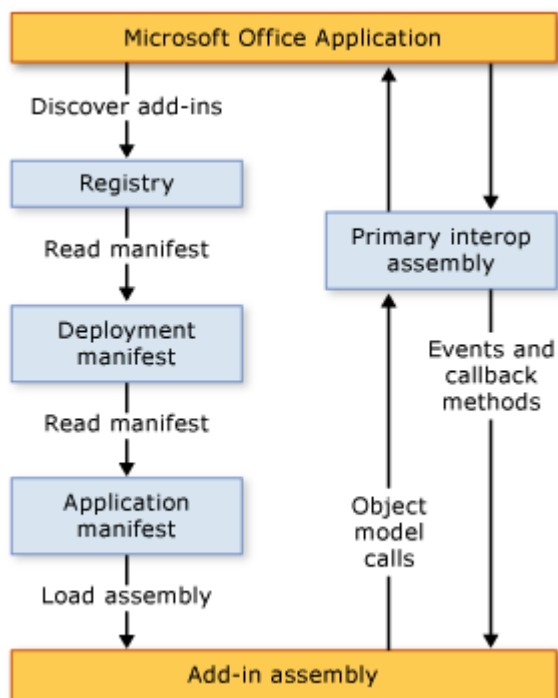
Doplněk používá oba manifesty k načtení a identifikaci poslední verze svého sestavení. Manifest rozložení (deployment manifest) odkazuje na aktuální aplikační manifest. Ten zase odkazuje na aktuální sestavení doplňku (knihovnu doplňku). Jedná se v podstatě o XML soubor obsahující informace o daném doplňku. Mimo jiné obsahuje informace o třídě, kterou se doplněk spouští (tzv. entry class) apod. Aplikační manifest je tvořen automaticky během procesu sestavení.

### 3.4.3 Visual Studio Tools for Office Runtime

Ke spuštění doplňku vytvořeného pomocí Office developer tools v MS Visual Studiu na koncovém počítači uživatele je zapotřebí mít nainstalované VSTO Runtime na tomto počítači. VSTO Runtime obsahuje komponenty a sadu knihoven. Komponenty slouží převážně pro zavedení doplňku do aplikace, knihovny pak poskytují objektový model, využívaný doplňkem, k rozšíření původní aplikace.

Předchozí verze VSTO runtime 8.0 používala pro zavedení doplňku knihovnu AddInLoader.dll – v registrech se pak uváděl odkaz na manifest doplňku (např. doplnek.dll.manifest). Novější verze 9.0 již tuto knihovnu neobsahuje a používá k zavedení doplňku knihovnu VSTOloader.dll (viz další kapitola). V registrech se pak také doporučuje uvádět odkaz na soubor typu \*.vsto daného doplňku. Pokud ještě není doplněk nainstalován, použije VSTO Runtime automaticky soubor VSTOInstaller.exe a doplněk se pokusí nainstalovat.

### 3.4.4 Spolupráce doplňku a MS Office aplikací



Obr. 4: Architektura doplňku[4]

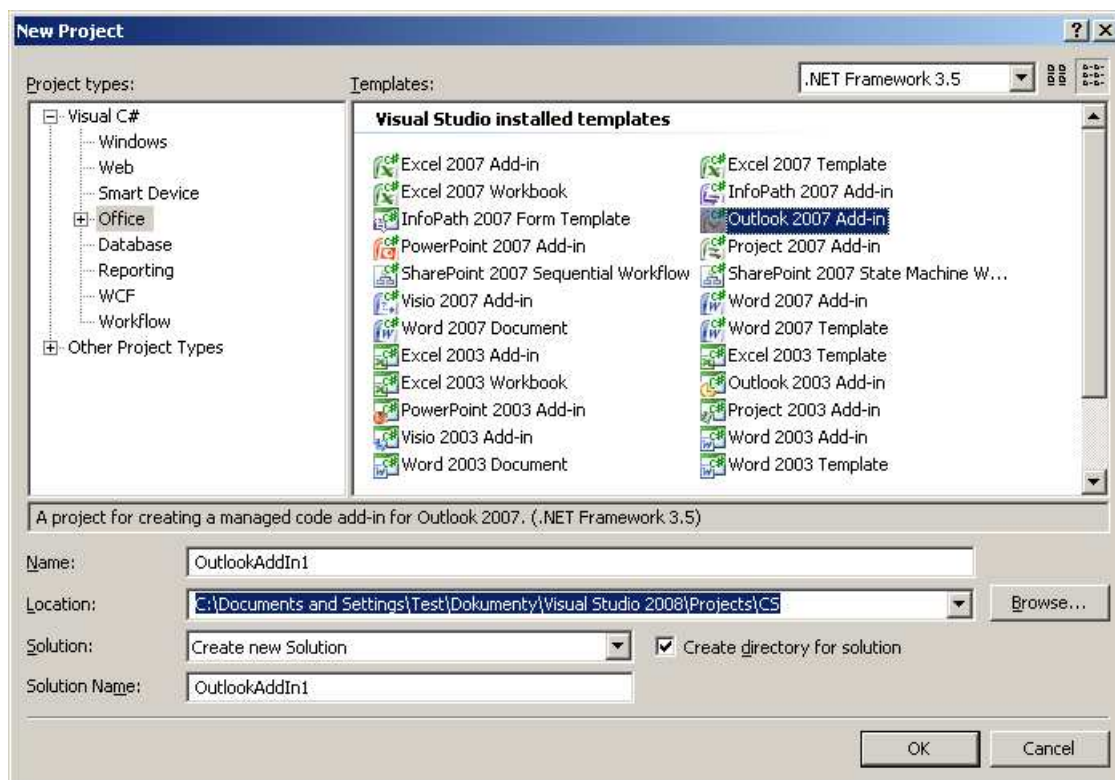
Při načítání doplňků během spuštění aplikace je postupováno následujícím způsobem:

- Aplikace ověří příslušné registry, zdali se zde nenachází registrované doplňky pro danou aplikaci.
- Najde-li aplikace registrované doplňky, načte knihovnu VSTOEE.dll, která načte knihovnu VSTOLoader.dll. Tyto knihovny jsou tzv. načítacími komponentami (loadery) pro VSTO runtime
- knihovna VSTOLoader.dll natáhne .NET Framework a spustí managed část VSTO runtime
- VSTO runtime ověří aktualizace manifestů a načte nejaktuálnější manifest rozložení a aplikační manifest
- VSTO runtime také spustí sérii bezpečnostních testů, a je-li aplikace označena jako důvěryhodná, pak VSTO runtime ověří pomocí manifestů poslední aktualizace daného doplňku. Je-li přítomná novější verze, pak ji VSTO runtime načte do „ClickOne“ cache klientského počítače
- Poté je vytvořena nová aplikační doména, do které je doplněk nahrán

- VSTO runtime se snaží zavolat metodu RequestComAddInAutomationService, je-li v doplňku implementována (přetížena = overridevaná)
- VSTO runtime se snaží zavolat metodu „RequestService“, je-li v doplňku implementována (přetížena = overridevaná)
- VSTO runtime volá metodu „ThisAddIn\_Startup“ v doplňku. Tato metoda je defaultní pro zpracování události „Startup“

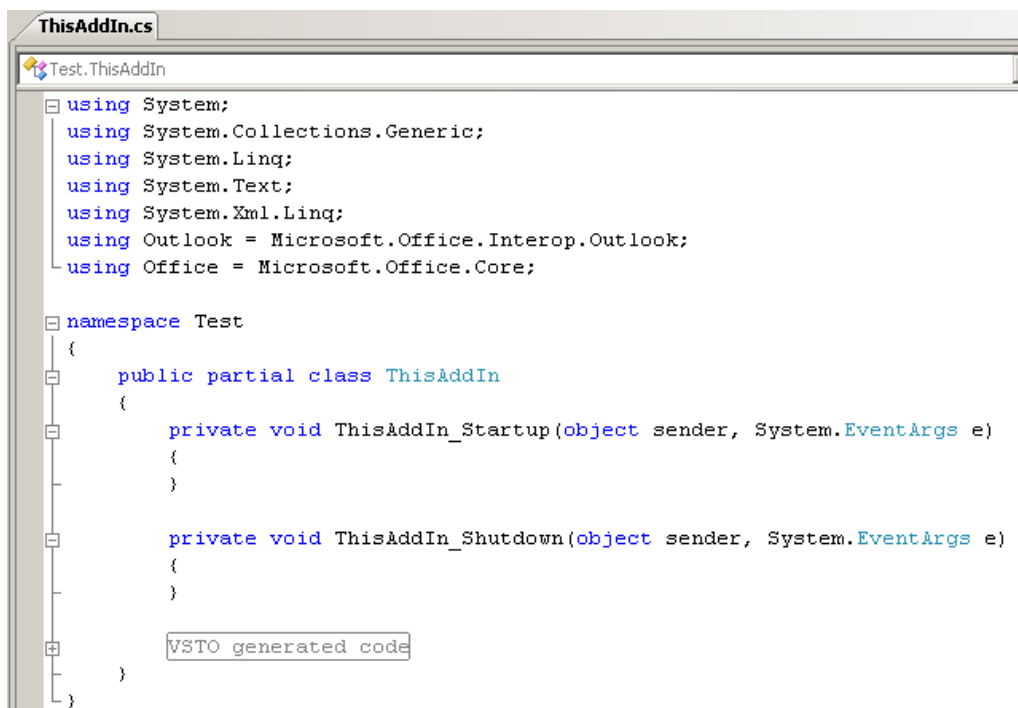
### 3.5 Vytvoření doplňku pro MS Office

Visual Studio s instalovanými nástroji pro vývoj doplňku pro MS Office nabízí několik šablon doplňků pro různé MS Office aplikace:



**Obr. 5: Šablony v MS Visual Studio 2008 pro vývoj doplňků pro MS Office**

Výběrem aplikace, pro kterou chceme daný doplněk vytvořit, zadáním názvu doplňku a místa jeho uložení a následným potvrzením dialogu založíme nový projekt pro realizaci doplňku.



**Obr. 6: ThisAddIn.cs - vstupní třída doplňku**

MS Visual Studio vygeneruje třídu „ThisAddIn.cs“, která obsahuje metody:

- *ThisAddIn\_Startup* – akce, které se mají provést při startu aplikace
- *ThisAddIn\_ShutDown* – akce, které se mají provést při ukončení aplikace

Doplněním kódu do těl těchto metod můžeme započít vývoj doplňku.

Do nově vzniklého projektu lze také přidávat další prvky jako nové třídy, různé controls, apod.

Co se týče GUI aplikace MS Outlook 2007, jsou zde dostupné tyto prvky:

- *Outlook Form Region* – představuje rozšíření pro okna reprezentována objektem Inspector (viz 3.5.1 Objektový model aplikace Outlook)
- *Ribbon a Ribbon (XML)* – představuje nový typ nástrojové lišty poprvé představené v MS Office 2007. Konfigurace lze provádět pomocí visual designeru MS Visual Studio nebo pomocí XML souboru (konfigurace komponent je zde obdobná jako v ASP.NET) .

Inicializace těchto prvků je před vlastním voláním metody „ThisAddIn\_Startup“

### 3.5.1 Objektový model aplikace Outlook

Objektový model Outlooku poskytuje třídy a rozhraní, které reprezentují jednotlivé položky v uživatelském rozhraní aplikace. Při vývoji doplňku pro aplikaci MS Outlook můžeme přicházet do styku s následujícími objekty:

- *Application* – je nejvýše položeným objektem ve struktuře objektového modelu Outlook. Představuje vlastní aplikaci MS Outlook
- *Explorer* – představuje okno, které zobrazuje položky obsažené ve vybrané složce (složka může obsahovat emaily, úkoly nebo připomínky)
- *Inspector* – představuje okno, které zobrazuje jednu položku (jako např. email, úkol, atd.)
- *MAPIFolder* – představuje složky, ve kterých jsou uloženy položky daného (shodného) typu (emaily, úkoly, připomínky, apod.)
- *MailItem* – představuje email (přijatý email, odeslaný email)
- *AppointmentItem* – představuje jednorázovou a opakovanou připomínku, připomínku v kalendáři, meeting
- *TaskItem* – představuje úkol, realizovatelný v daném časovém rozmezí
- *ContactItem* – představuje kontakt ve složce kontaktů

## 3.6 Způsoby instalace doplňku

Pro instalaci doplňků (Add-In) aplikací MS Office 2007 v prostředí MS Visual Studio 2008 lze využít dva způsoby:

- pomocí vestavěné funkce nazvané „ClickOne“
- pomocí Windows Installeru – „Setup“ projekt

### 3.6.1 Instalátor „ClickOne“

Tímto způsobem lze maximálně zjednodušit vytvoření instalačního balíčku. Stačí pouze určit, kde se má instalační balíček vytvořit, kde má popřípadě hledat aktualizace (samozřejmě je podpora sítí, internetu i pevných disků a médií jako CD/ DVD), a instalační balíček se automaticky vytvoří. Jednoduchost tohoto řešení má ale řadu nevýhod. Jako jednu z hlavních můžeme uvést nemožnost jakkoli ovlivnit vlastní proces instalace (vlastností této

instalace je i nemožnost určit, kde se bude doplněk instalovat – všechny registrace doplňku se budou např. odkazovat na místo, odkud byl instalátor spuštěn), rovněž nelze přidávat další zdroje (jako podpůrné konfigurační a databázové soubory apod.), atd.

### **3.6.2 Instalátor „Windows Installer“**

Tento způsob vytváření instalačního balíčku je mnohem komplikovanější oproti předešlému. Nemá tolik funkcí automatizovaných, jako „ClickOne“ instalace, na druhou stranu má více možností jak instalaci ovlivnit a upravit aktuálním potřebám aplikace (v tomto případě doplňku) pro který instalátor vzniká.

Pro správnou instalaci doplňku, je nutno provést adekvátní záznamy v registrech (viz 3.4.1 Registrace doplňku) a přidat manifest a vsto soubory doplňku (tyto nejsou do instalačního projektu automaticky přidány pomocí funkce „Add/ Project Output“ – touto funkcí je přidána pouze hlavní knihovna \*.dll doplňku)

Windows instalátor umožňuje řídit vlastní průběh instalace prostřednictvím vestavěných „setup“ dialogů, umožňuje přidání dalších podpůrných souborů a adresářů, včetně systémových. Rovněž je zde možnost definovat vlastní požadavky na systémové prostředky pro spuštění instalace (tuto možnost má samozřejmě i instalace typu „ClickOne“, ale je zde opět omezená na výběr z několika modulů, které jsou pro správný průběh instalace a funkce instalované aplikace-doplňku vyžadovány).

Instalace pomocí Windows instalátoru má rovněž možnost definovat uživatelské akce, které se mají vykonat při instalaci, po instalaci, při odinstalaci apod.

Na závěr je potřeba podotknout, že i pokud použijeme Windows instalátor, konečná fyzická registrace doplňku bude nakonec provedena pomocí zmiňovaného „ClickOne“ instalátoru (zajištěno pomocí VSTO Runtime) a to při prvním spuštění dané MS Office aplikace pro kterou byl doplněk instalován.



## 4 Implementace GTD do prostředí MS

### Outlook

Tato část práce si klade za úkol zhodnotit aplikaci MS Outlook verze 2007 z hlediska možností využívání principů metodologie GTD. Rovněž si klade za cíl analýzu a návrh implementace těch funkcionalit, vyžadovaných metodologií GTD, které v MS Outlook 2007 chybí.

#### 4.1 Analýza implementace GTD

##### 4.1.1 Analýza MS Outlooku z hlediska GTD

Aplikace MS Outlook je plnohodnotnou PIM aplikací. Obsahuje e-mailového klienta, kalendáře, poznámky, správu kontaktů a další podružné funkcionality, nezbytné k efektivní organizaci našeho času.

Přestože je tato aplikace poměrně robustní a disponuje opravdu sofistikovanými time managementovými funkcemi, chybí zde pokročilejší funkce na správu záležitostí, jakými jsou emaily, připomínky v kalendáři či úkoly. Tyto záležitosti jdou samozřejmě v MS Outlooku efektivně zpracovávat a pro základní používání většině uživatelů standardní funkčnost aplikace postačuje.

Pokud však zpracováváme větší množství záležitostí, je nutné z hlediska přehlednosti, vytvořit systém, který dokáže zorganizovat související záležitosti a uspořádat je do smysluplných celků. Aktuálně nám MS Outlook poskytuje možnost řazení zpracovávaných záležitostí do kategorií. Tyto lze libovolně modifikovat, přidávat apod. Dle těchto kategorií pak lze jednotlivé položky řadit a vyhledávat, tudíž můžeme jednoduchým způsobem získat obecný přehled. Nicméně v případě většího počtu záležitostí zařazených do klasických kategorií MS Outlook se přehlednost výrazně snižuje.

Dalším aspektem, ve kterém má MS Outlook rezervy, je úplná absence projektů. Lze namítnout, že pro projektové plánování je taky produkt MS Project. Bohužel ne každý, kdo vlastní aplikaci MS Outlook disponuje také tímto řešením. Navíc je produkt MS Project postaven jako aplikace pro podporu projektového řízení, zdrojů, úkolů apod. Je více projekt

managementovým než time managementovým nástrojem. V neposlední řadě je MS Project samostatnou aplikací, a i když bývá často prezentován jako součást MS Office, tak nikdy nebyl v žádné oficiálně vydané verzi MS Office přítomen.

Obdobným způsobem MS Outlook 2007 postrádá systém archivace záležitostí, které vyžadujeme ponechat z důvodů důležité informace nebo pozdějšího zvážení. Tímto není myšlena složka archívu, která je v MS Outlooku přítomna a slouží k archivaci - kompresi starých a dlouho nepoužívaných záležitostí (hlavně položky odeslané pošty)

Tyto tři výše zmíněné nedostatky, vyžadované metodologií GTD, se nyní pokusíme analyzovat a posléze implementovat do prostředí MS Outlook 2007.

## **4.1.2 Specifikace požadavků pro implementaci GTD**

### **4.1.2.1 Obecné předpoklady, specifikace vstupů a výstupů**

Požadovaný doplněk (Add-in) nebude měnit stávající funkčnost aplikace MS Outlook, ani nijak přímo zasahovat do stávajícího prostředí způsobem, který by uživatele nutil používat aplikaci jiným způsobem, nebo znemožňoval ji využívat způsobem, na jaký byl doposud zvyklý.

Veškeré nové funkčnosti aplikované pomocí doplňku budou rozšiřovat funkčnost aplikace MS Outlook v daném, dále popsaném, rozsahu, aniž by omezovaly standardní využívání aplikace.

Novými vstupy, které budou realizaci doplňku přidány do aplikace MS Outlook jsou:

- Seznam rolí, které budou seskupovat kategorie a projekty
- Seznam GTD kategorií (nezávislé na kategoriích implementovaných v MS Outlook)
- Seznam projektů
- Seznam GTD archívních složek

Doplňek by měl prostřednictvím MS Outlook poskytovat tyto výstupy:

- Přehled objektů MS Outlook dle jejich závislosti na GTD kategoriích
- Přehled objektů MS Outlook dle jejich v závislosti na projektech
- Přehled objektů MS Outlook dle jejich závislosti na GTD archívech

#### 4.1.2.2 Kategorie závislé na roli uživatele

Pro umožnění rychlého výběru záležitostí, kterým se v dané chvíli chceme věnovat, nám slouží kategorie. Kategorie tak, jak jsou implementovány v MS Outlooku, ztrácí svoji účinnost při větším počtu kategorizovaných záležitostí a to z důvodu chybějící možnosti kategorie hierarchicky uspořádat.

Doplňěk ponechá stávající úlohu Outlook kategorií nedotčenou a bude implementovat vlastní „GTD“ kategorie, které budou umožňovat hierarchické řazení, dle požadavků uživatele. Kategorie bude reprezentována jménem a případným detailním popisem.

Ještě efektivnější způsob kategorizace záležitostí, je možnost zařazení kategorií a sub-kategorií do uživatelských rolí. Uživatel zde má právo, zvolit si roli v jaké bude aktuálně vystupovat (např. student, zaměstnanec, rodič, atp.) a dle zvolené role jsou mu poté nabídnuty kategorie (a sub-kategorie) příslušející dané roli.

#### 4.1.2.3 Projekty závislé na roli uživatele

Pokud zpracovávaná záležitost je rozsáhlejšího rázu a nelze ji vyřídit jedním krokem dle zásad GTD, je zapotřebí tyto další kroky specifikovat a zároveň je z důvodu přehlednosti udržovat pohromadě. Tuto funkčnost budou v rámci doplňku zajišťovat projekty, které v MS Outlooku úplně chybí.

Projekty budou implementovány na obdobném principu jako kategorie. Rovněž bude možnost jejich hierarchického uspořádání a také budou závislé na roli, v jaké bude uživatel aktuálně vystupovat. Dle zvolené role pak získá nabídku odpovídajících projektů.

Každý projekt bude obsahovat:

- název projektu,
- detailní popis projektu,
- stav rozpracovanosti projektu (0...100%)
- stav, ve kterém se projekt nachází:
  - o *K posouzení* = sbírání podkladů, analýza požadavků apod.
  - o *K vypracování* = projekt se bude realizovat
  - o *Rozpracován* = projekt je aktuálně realizován
  - o *Delegován* = projekt čeká nebo byl předán třetí osobě (stále je ale v naší kompetenci)
  - o *Dokončen* = projekt je hotov

- *Zrušen* = projekt byl v jakékoli předchozí fázi (kromě „dokončen“) zrušen
- prioritu projektu:
  - nízká
  - normální
  - vysoká
- datum odevzdání (dokončení) projektu

#### **4.1.2.4 Archívy**

Archívy budou sloužit pro uchování (zatřídění) věcí, o kterých jsme přesvědčení, že bychom je mohli v budoucnu použít.

Funkčně budou archívy implementovány obdobným způsobem jako kategorie nebo projekty. Tedy bude se jednat o hierarchickou strukturu složek, do kterých bude možnost ukládat dané záležitosti.

Nicméně u archívu nebude závislost na roli, v jaké uživatel v daném okamžiku vůči systému vystupuje. Nezávislost archívu na roli, bude výhodou v případě vyhledávání obecných informací, jelikož nelze deklarovat, že informace uložené do archívu v rámci jedné role, nebudou nikdy uživatelem vyžadovány v rámci roli jiné. Závislost archívů na rolích by pak koncového uživatele mohla výrazně omezovat.

#### **4.1.2.5 Omezení hierarchie kategorií, projektů a archívů**

Záležitosti půjdou v rámci hierarchie výše zmíněných entit (míněno kategorie, projekty a archívy) ukládat pouze do entit, které nejsou rodiči, tudíž neobsahují další entity (potomky) stejného typu. Např. nelze objektu (mail, úkol, připomínka) přiřadit kategorií, která obsahuje další kategorií (kategorie). Objektu půjde přiřadit pouze tzv. „koncovou“ kategorií, tj. kategorií, která je v hierarchickém stromu kategorií pomyslným listem.

Účelem tohoto omezení je hlavně udržet přehlednost ve stromech daných entit. Tak jak to vyžaduje i metodologie GTD, aby specifikace dalších kroků byla konkrétní, nikoli obecná, je tento samý přístup kladen také na uspořádání a specifikace kategorií, projektů a archívů. Tudíž je vždy nutno zařadit daný objekt do koncové kategorie (v případě kategorií), která představuje konkrétní specifikaci kategorie a nikoli do kategorie rodičovské, která představuje jakýsi obecný kontejner pro další, již upřesněné, kategorie.

#### **4.1.2.6 Outlook objekty, pravidla pro jejich zařazování**

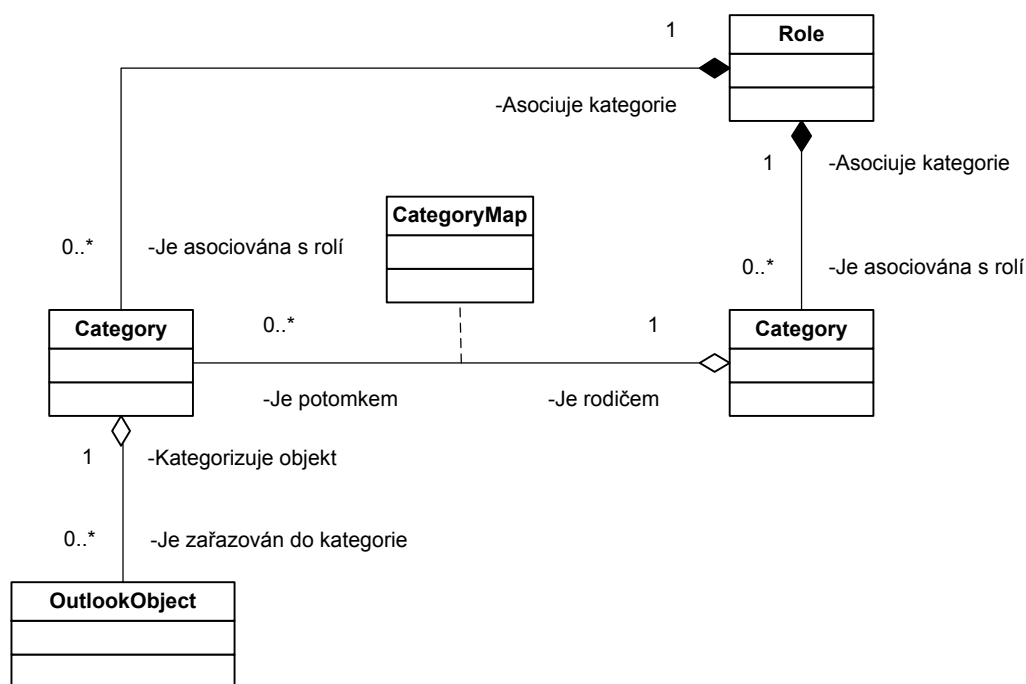
Výše definované kategorie, projekty a archívy budou sloužit k přehlednějšímu zařazení objektů aplikace MS Outlook. Toto řazení se bude týkat objektů:

- Přijatý (došlý) email
- Připomínka v kalendáři
- Úkol

Každému objektu tohoto typu bude možno přiřadit GTD kategorii, projekt nebo archív. Z předchozích zmíněných se navzájem dle pravidel metodologie GTD vylučují archív (tj. v terminologii GTD „Někdy – možná“) a projekt + kategorie. Tzn., že objekt může mít přiřazenu kategorii a současně také projekt, ale už nikoli archív. Obdobně pokud objekt zařadíme do archívu, pak již nelze objekt zařadit ani do kategorie, ani do projektu.

### 4.1.3 Konceptuální analýza

#### 4.1.3.1 Správa kategorií



Obr. 7: Třídní diagram - správa kategorií

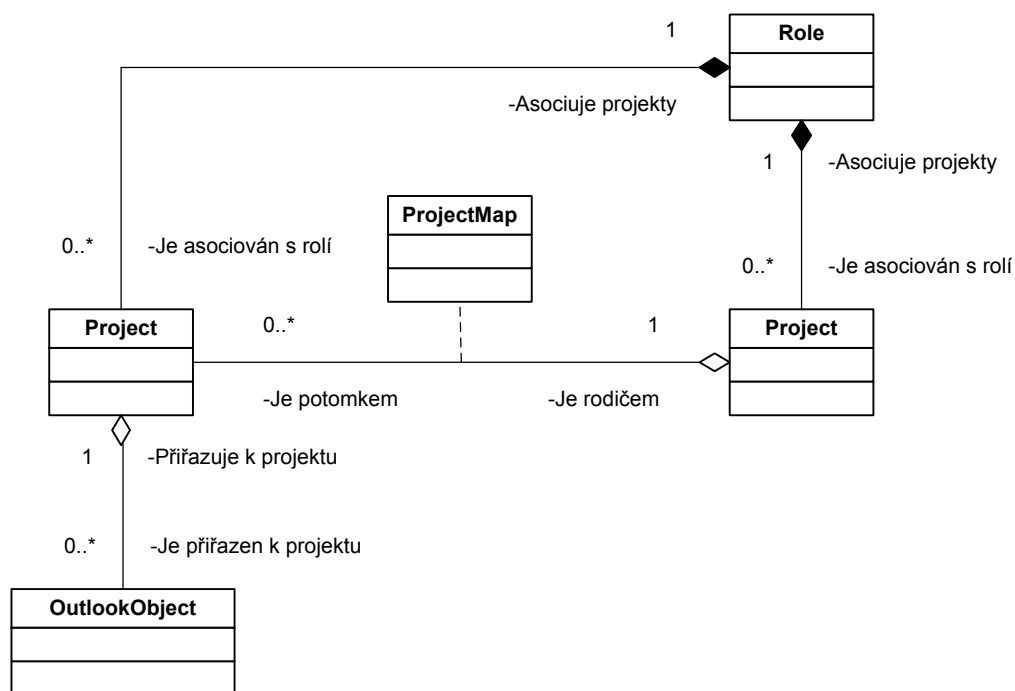
Každá instance entity „Category“ může (není podmínkou) mít jako potomky další instance entity „Category“. Jednotlivé instance entitního typu „Category“ pak mohou díky vzájemným vazbám vytvořit hierarchii (strom) instancí typu „Category“. Vzájemné mapování mezi dvěma instancemi typu „Category“ je zajišťováno vazební entitou typu „CategoryMap“.

Každá instance typu „Category“ je povinně mapována na instanci entitního typu „Role“.

Každé instanci entitního typu „OutlookObject“, představující (mapující) konkrétní objekt v aplikaci MS Outlook, může být přiřazena libovolná instance typu „Category“, s omezením, že:

- Na danou instanci typu „Category“ není vázána žádná další instance typu „Category“ (viz. 4.1.2.5 Omezení hierarchie kategorií, projektů a archívů).
- Instanci typu „OutlookObject“ není přiřazena žádná instance entitního typu „Archive“ (viz. 4.1.2.6 Outlook objekty, pravidla pro jejich zařazování).

#### 4.1.3.2 Správa projektů



Obr. 8: Třídní diagram - správa projektů

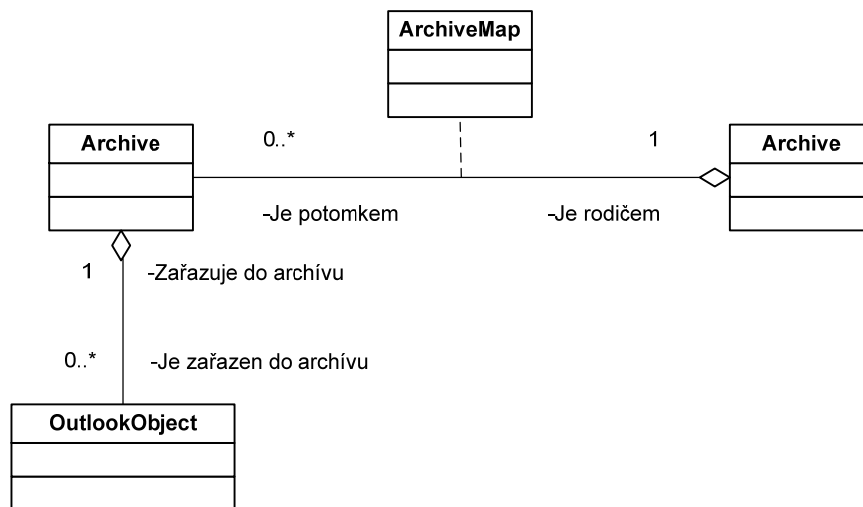
Správa projektů je už dle zobrazení shodná se správou kategorií. Je zde nově entitní typ „Project“ jehož instance na sebe mohou navazovat prostřednictvím vazební entity „ProjectMap“ a tvořit tak stromovou strukturu instancí typu „Project“.

Každá instance typu „Project“ je opět povinně vázána k instanci typu „Role“.

Instance typu „OutlookObject“ může být mapována na instanci typu „Project“ za obdobných předpokladů jako v případě kategorií, tj.:

- Na danou instanci typu „Project“ není vázána žádná další instance typu „Project“.
- Instanci typu OutlookObject není aktuálně přiřazena žádná instance typu „Archive“.

#### 4.1.3.3 Správa archívů



Obr. 9: Třídní diagram - správa archívů

Instance typu „Archive“ již nemají pravidlo pevné vazby na instanci typu „Role“ jak tomu bylo u kategorií nebo u projektů.

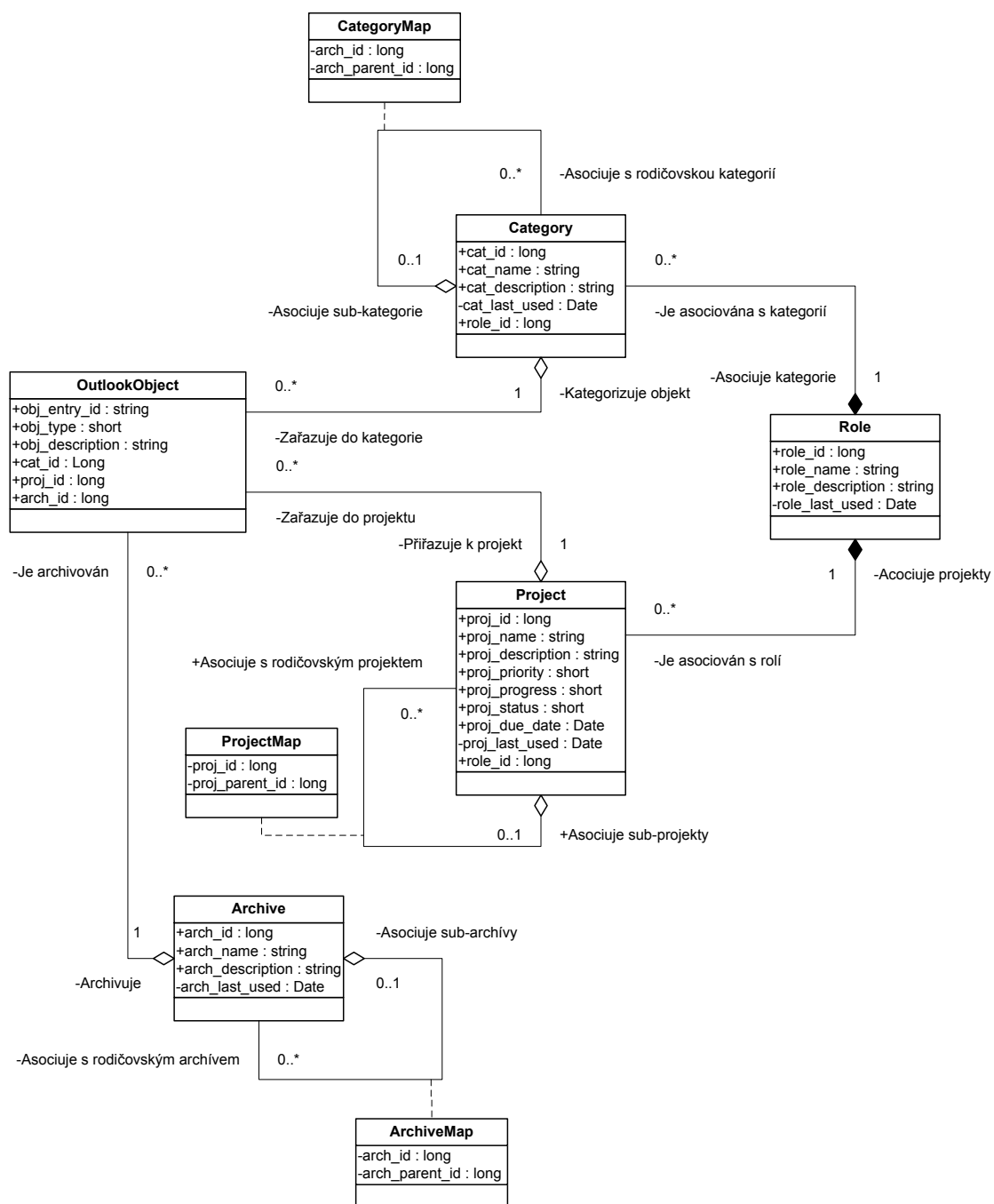
Každá instance typu „Archive“ může být opět mapována na jinou instanci typu „Archive“ prostřednictvím vazební entity „ArchiveMap“ a vytvářet tak hierarchii archívů obdobně jako u kategorií nebo projektů.

Instance typu „OutlookObject“ pak může být mapována na instanci typu „Archive“ za předpokladu, že.:

- Na danou instanci typu „Archive“ není vázána žádná další instance typu „Archive“.
- Instanci typu OutlookObject není aktuálně přiřazena žádná instance typu „Category“ nebo „Project“.



#### 4.1.4 Databáze



Obr. 10: Konceptuální model databáze

#### 4.1.4.1 Lineární zápis

**Role** (role\_id, role\_name, role\_description, role\_last\_used)

**Category** (cat\_id, cat\_name, cat\_description, cat\_last\_used, *role\_id*)

**CategoryMap** (cat\_id, cat\_parent\_id)

**Project** (proj\_id, proj\_name, proj\_description, proj\_status, proj\_priority, proj\_progress, proj\_due\_date, proj\_last\_used, *role\_id*)

**ProjectMap** (proj\_id, proj\_parent\_id)

**Archive** (arch\_id, arch\_name, arch\_description, arch\_last\_used)

**ArchiveMap** (arch\_id, arch\_parent\_id)

**OutlookObject** (obj\_entry\_id, obj\_type, obj\_description, *cat\_id*, *proj\_id*, *arch\_id*)

#### 4.1.5 Datová analýza

**Tabulka 2: Datový slovník**

entita	atribut	dat. typ	KEY	NULL	IDX	poznámka
Role	role_id	long	A	-	A	
	role_name	char(30)	-	-	-	
	role_description	char(100)	-	A	-	
	role_last_used	date	-	A	-	
Category	cat_id	long	A	-	A	
	cat_name	char(30)	-	-	-	
	cat_description	char(100)	-	A	-	
	cat_lat_used	date	-	A	-	
	role_id	long	-	-	A	FK
CategoryMap	cat_id	long	A	-	A	FK
	cat_parent_id	long	A	-	A	FK
Project	proj_id	long	A	-	A	
	proj_name	char(30)	-	-	-	
	proj_description	char(100)	-	A	-	
	proj_progress	short	-	-	-	0-100%

entita	atribut	dat. typ	KEY	NULL	IDX	poznámka
Project	proj_priority	short	-	-	-	priorita <sup>1</sup>
	proj_status	short	-	-	-	stav projektu <sup>2</sup>
	proj_due_date	date	-	-	-	
	proj_last_used	date	-	A	-	
	role_id	long	A	-	A	FK
ProjectMap	proj_id	long	A	-	A	FK
	proj_parent_id	long	A	-	A	FK (proj_id)
Archive	arch_id	long	A	-	A	
	arch_name	char(30)	-	-	-	
	arch_description	char(100)	-	A	-	
	arch_last_used	date		A	-	
ArchiveMap	arch_id	long	A	-	A	FK
	arch_parent_id	long	A	-	A	FK (arch_id)
OutlookObject	obj_entry_id	char(255)	A	-	A	
	obj_type	short	-	-	-	typ <sup>3</sup>
	obj_description	char(100)	-	A	-	
	cat_id	long	-	A	A	FK
	proj_id	long	-	A	A	FK
	arch_id	long	-	A	A	FK

---

<sup>1</sup> Použitá enumerace 0=Nízká, 1=Normální, 2=Vysoká

<sup>2</sup> Použitá enumerace 0=K posouzení, 1=K vypracování, 2=Rozpracován, 3=Delegován, 4=Dokončen, 5=Zrušen

<sup>3</sup> Použitá enumerace 0=Email, 1=Připomínka, 2=Úkol

## 4.1.6 Funkční analýza

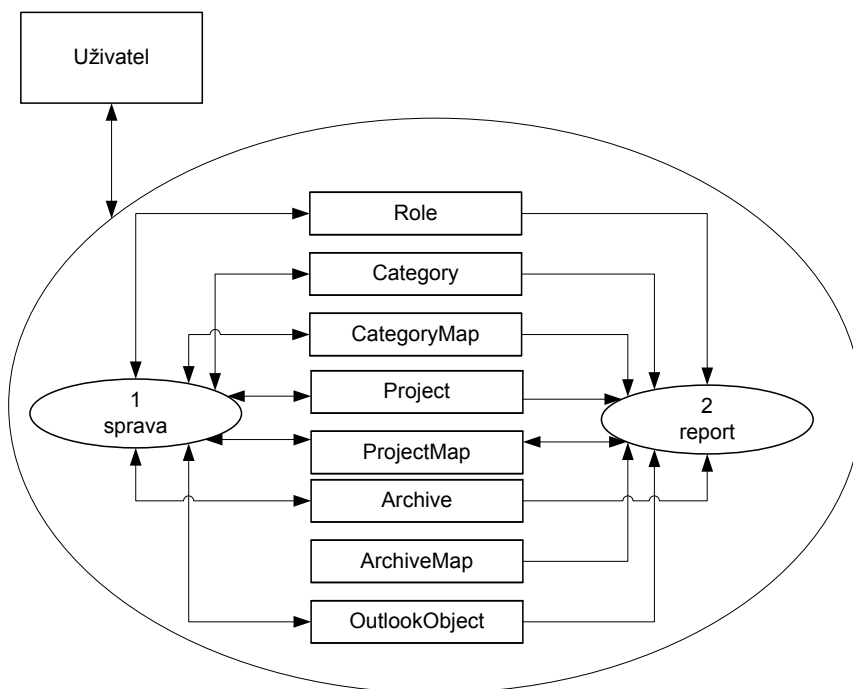
### 4.1.6.1 Seznam elementárních funkcí

**Tabulka 3: Funkční analýza-seznam elementárních funkcí**

I. Subsystem	II. Subsystem	III. Subsystem	Popis funkce
1 sprava	1.1 role_sprava	1.1.1 role_nova	vytvoření nové role
		1.1.2 role_edit	úprava stávající role
		1.1.3 role_odstran	odstranění role
	1.2 kat_sprava	1.2.1 kat_nova	vytvoření nové kategorie
		1.2.2 kat_edit	úprava stávající kategorie
		1.2.3 kat_odstran	odstranění kategorie
	1.3 proj_sprava	1.3.1 proj_novy	vytvoření nového projektu
		1.3.2 proj_edit	úprava stávajícího projektu
		1.3.3 proj_odstran	odstranění projektu
	1.4 arch_sprava	1.4.1 arch_novy	vytvoření nového archívu
		1.4.2 arch_edit	úprava stávajícího archívu
		1.4.3 arch_odstran	odstranění archívu
	1.5 obj_sprava	1.5.1 obj_kategorie	zařazení objektu aplikace Outlook do kategorie
		1.5.2 obj_projekt	zařazení objektu aplikace Outlook k projektu
		1.5.3 obj_archiv	zařazení objektu aplikace Outlook do archívu
2 report	2.1 rep_kategorie	-	zobrazení objektů aplikace Outlook dle jejich zařazení v kategoriích
	2.2 rep_projekt	-	zobrazení objektů aplikace Outlook dle jejich příslušnosti k projektům
	2.2 rep_archiv	-	zobrazení objektů aplikace Outlook dle jejich zařazení v archívech

#### 4.1.6.2 Rozdělení funkcí do subsystému (Data Flow Diagramy)

##### 4.1.6.2.1 DFD 1. úrovně

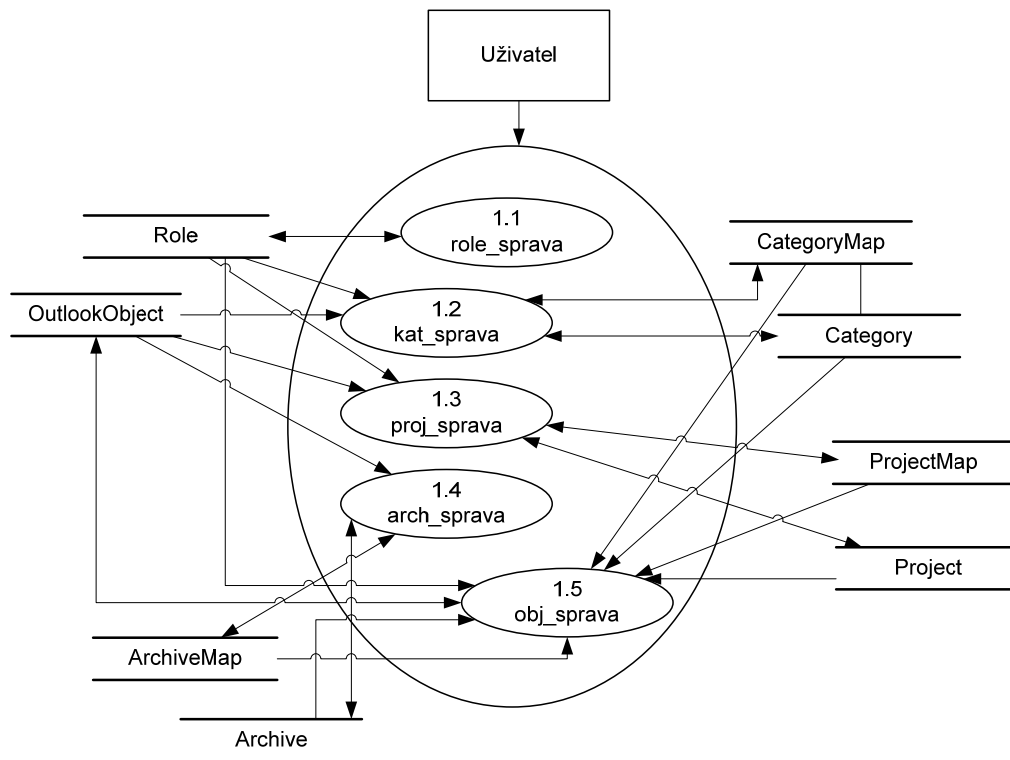


Obr. 11: DFD 1. řádu (subsystémy „1 sprava“ a „2 report“)

Hlavní systém se dělí na dva subsystémy:

- Správa – tento subsystém zajišťuje administraci GTD kategorií, projektů a archívů (tj. jejich vytváření, změny, zrušení), a dále zajišťuje funkčnost samotného řazení objektů aplikace MS Outlook do výše zmíněných kategorií, projektů a archívů
- Report – tento subsystém zajišťuje vlastní reportování – tj. podávání přehledných informací uživateli o záležitostech nastavených v subsystému „Správa“ (jedná se hlavně o informace, jak jsou jednotlivé objekty aplikace Outlook kategorizovány, archivovány nebo uloženy v archívech)

#### 4.1.6.2.2 DFD 2. úrovně – funkce subsystému „1. Správa“



Obr. 12: DFD 2. řádu (subsystémy náležející do 1. subsystému „1 správa“)

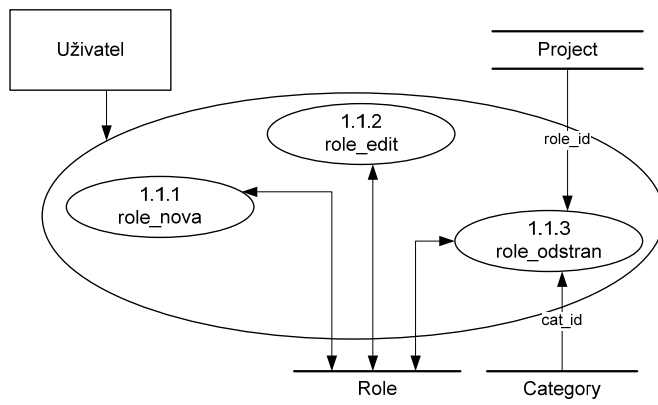
Subsystémy 2. úrovně v subsystému „Správa“ zajišťují vytváření, editaci a odstranění pro tyto entity:

- Role
- Kategorie
- Projekt
- Archív

Dále pak zajišťují mapování objektů aplikace MS Outlook to instancí těchto entit.

Jednotlivé subsystémy jsou uvedeny níže, včetně minispecifikací, které upřesňují funkčnost jednotlivých subsystému na 3. úrovni DFD.

#### 4.1.6.2.3 DFD 3. úrovně – funkce subsystému „1.1 role\_sprava“



Obr. 13: DFD 3. řádu (subsystémy náležející do 2. subsystému „role\_sprava“)

##### 4.1.6.2.3.1 Návrh algoritmu subsystému „1.1.1 role\_nova“

1. Uživatel otevře buďto „task pane“ pro reporty nebo objekt aplikace MS Outlook, na který se vztahuje možnost zařazení do systému GTD (došlý email, připomínka v kalendáři a úkol).
2. Zobraz dle předchozího výběru uživatele příslušnou lištu nástrojů, která obsahuje tlačítka pro administraci kategorií a projektů (administrace rolí nebude implementována samostatně, ale bude součástí implementace administrace kategorií a projektů, které jsou na rolích závislé).
3. Uživatel kliknutím na jednu z výše zmíněných tlačítek v liště nástrojů otevře dialogové okno pro správu kategorií nebo projektů.
4. Zobraz dialogové okno, které bude obsahovat nahoře lištu tlačítek a v ní tlačítka pro správu rolí (tlačítka „Přidat roli“, „Upravit roli“ nebo „Odstranit roli“).
5. Pod lištou nabídek se nachází komponenta pro vykreslení stromu. Zobraz stromovou strukturu kategorií (nebo projektů – dle volby v bodě 3) v závislosti na rolích, tak, aby role byly jako kořeny a projekty (kategorie) pak navazovaly přímo na ně – funkce pro konstrukci a zobrazení stromu bude blíže specifikována v subsystému „2.1 rep\_kategorie“ a „2.2 rep\_projekt“.
6. Uživatel klikne na tlačítko „Přidat roli“.
7. Načti z relace „Role“ všechna jména dosud použitá v existujících rolích. Otevři dialog pro přidání role, předej načtená jména do tohoto dialogu a požaduj zadání

jména a popisu role. Dialog se potvrdí kliknutím na tlačítko „OK“, stornuje se tlačítkem „Storno“.

8. Klikne-li uživatel na tlačítko „OK“, pokračuj dalším bodem. Jinak uzavři dialog a ukonči funkci.
9. Ověř, zdali jméno role je jedinečné, zdali je zadáno a zdali nezačíná mezerou, popis není nutno zadávat. Je-li některá z předešlých kontrol pozitivní, pak upozorni uživatele na daný problém a požaduj opravu zadání. Jinak pokračuj dále.
10. Vytvoř nový záznam v relaci „Role“ se jménem a popisem, který byl zadán prostřednictvím dialogu.
11. Proveď aktualizaci stromu a ukonči funkci.

#### **4.1.6.2.3.2 Návrh algoritmu subsystému „1.1.2 role\_edit“**

1. První kroky jsou shodné s kroky 1 – 5 z předešlé části (viz. 4.1.6.2.3.1 Návrh algoritmu subsystému „1.1.1 role\_nova“).
2. Uživatel vybere ze zobrazeného stromu jednu roli a klikne na tlačítko „Upravit roli“
3. Načti z relace „Role“ všechny jména dosud použitá v existujících rolích, kromě jména aktuálně editované role. Otevři dialog pro úpravu role, předej načtená jména do tohoto dialogu, vyplň textová pole se jménem a popisem, daty získanými z aktuálně vybrané role. Umožni změnu jména a popisu role. Dialog se potvrdí kliknutím na tlačítko „OK“, stornuje se tlačítkem „Storno“.
4. Ověř, zdali jméno role je jedinečné, zdali je zadáno a zdali nezačíná mezerou, popis není nutno zadávat. Je-li některá z předešlých kontrol pozitivní, pak upozorni uživatele na daný problém a požaduj opravu zadání. Jinak pokračuj dále.
5. Uprav záznam v relaci „Role“ s atributem role\_id = role\_id aktuálně upravované role. V záznamu změn jméno a popis dle hodnot zadaných prostřednictvím editačního dialogu.
6. Proveď aktualizaci stromu a ukonči funkci.

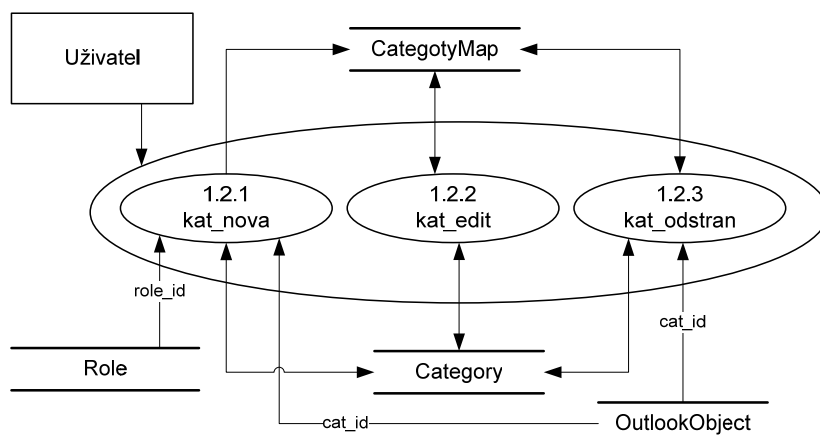
#### **4.1.6.2.3.3 Návrh algoritmu subsystému „1.1.3 role\_odstran“**

1. První kroky jsou shodné s kroky 1 – 5 z předešlé části (viz. 4.1.6.2.3.1 Návrh algoritmu subsystému „1.1.1 role\_nova“).
2. Uživatel vybere ze zobrazeného stromu jednu roli a klikne na tlačítko „Odstranit roli“.



3. Zobraz dotaz, zdali je si uživatel vědom toho, že daná role bude odstraněna. Jeli odpověď kladná, pokračuj dále. Jinak ukonči funkci.
4. Ověř, zdali se v relacích „Project“ a „Category“ nenachází záznam kde hodnota atributu „role\_id“ = hodnotě atributu „role\_id“ aktuálně vybrané role. Nenachází-li se pak odstraň ze relace „Role“ záznam s hodnotou atributu „role\_id“ = hodnotě „role\_id“ aktuálně vybrané role.
5. Proveď aktualizaci stromu a ukonči funkci.

#### 4.1.6.2.4 DFD 3. úrovně – funkce subsystému „1.2 kat\_sprava“



Obr. 14: DFD 3. řádu (subsystémy náležející do 2. subsystému „kat\_sprava“)

##### 4.1.6.2.4.1 Návrh algoritmu subsystému „1.2.1 kat\_nova“

1. Uživatel otevře buďto „task pane“ pro reporty nebo objekt aplikace MS Outlook, na který se vztahuje možnost zařazení do systému GTD (došlý email, připomínka v kalendáři a úkol).
2. Zobraz dle předchozího výběru uživatele příslušnou lištu nástrojů, která obsahuje mimo jiné tlačítko „Administrace rolí a kategorií“.
3. Uživatel kliknutím na toto tlačítko v liště nástrojů otevře dialogové okno pro správu kategorií.
4. Zobraz dialogové okno, které bude obsahovat nahoře lištu tlačítek a v ní tlačítka pro správu kategorií (tlačítka „Přidat kategorií“, „Upravit kategorií“ nebo „Odstranit kategorií“).

5. Pod lištou nabídek se nachází komponenta pro vykreslení stromu. Zobraz stromovou strukturu kategorií v závislosti na rolích, tak, aby role byly jako kořeny a na ně pak navazovaly příslušné kategorie (dle návaznosti dané relací „CategoryMap“) – funkce pro konstrukci a zobrazení stromu bude blíže specifikována v subsystému „2.1 rep\_kategorie“.
6. Uživatel vybere ze stromu buďto roli, nebo kategorii, která bude sloužit jako rodičovská pro aktuálně přidávanou. Pak klikne na tlačítko „Přidat kategorii“.
7. Pokud je jako rodič nově vznikající kategorie vybrána také kategorie, zkontroluj, zdali hodnota atributu „cat\_id“ rodičovské kategorie není obsažena v relaci „OutlookObject“ v atributu „cat\_id“. V případě že ano, oznam uživateli, že daná kategorie nemůže být použita jako rodič, jelikož obsahuje objekty MS Outlook, a ukonči vykonávání funkce
8. Načti z relace „Category“ všechny jména dosud použitá v rodičovské kategorii. Nemá-li vybraná kategorie rodiče, pak načti všechny jména kořenových („root“) kategorií. Rovněž získej hodnotu atributu „cat\_id“ a „role\_id“ z vybrané rodičovské kategorie nebo role (v případě role zůstává hodnota atributu „cat\_id“ prázdná). Otevři dialog pro přidání kategorie, předej dříve načtené hodnoty do tohoto dialogu a požaduj zadání jména a popisu kategorie. Dialog se potvrdí kliknutím na tlačítko „OK“, stornuje se tlačítkem „Storno“.
9. Klikne-li uživatel na tlačítko „OK“, pokračuj dalším bodem. Jinak uzavři dialog a ukonči funkci.
10. Ověř, zdali jméno kategorie je jedinečné v daném kontextu, zdali je zadáno a zdali nezačíná mezerou, popis není nutno zadávat. Je-li některá z předešlých kontrol pozitivní, pak upozorni uživatele na daný problém a požaduj opravu zadání. Jinak uzavři dialog pro vytvoření nové kategorie a pokračuj dále.
11. Vytvoř nový záznam v relaci „Category“ se jménem a popisem, který byl zadán prostřednictvím dialogu, doplň také atribut „role\_id“ dle vybraného rodiče. Byla-li jako rodič vybraná kategorie, pak získej hodnotu „cat\_id“ nově vytvořené kategorie a proved' zápis nového záznamu do relace „CategoryMap“, kde atribut „cat\_id“ je roven hodnotě atributu „cat\_id“ nově vytvořeného záznamu v relaci „Category“ a atribut „cat\_parent\_id“ = hodnotě atributu „cat\_id“ rodičovské kategorie. Zápis obou záznamů do relací „Category“ a „CategoryMap“ je nutno provést v rámci transakce (jde o dva na sobě závislé záznamy).

12. Proved' aktualizaci stromu a ukonči funkci.

#### **4.1.6.2.4.2 Návrh algoritmu subsystému „1.2.2 kat\_edit“**

1. První kroky jsou shodné s kroky 1 -5 popsány v předchozí kapitole (viz. 4.1.6.2.4.1 Návrh algoritmu subsystému „1.2.1 kat\_nova“).
2. Uživatel ze stromu vybere jednu ze zobrazených kategorií a klikne na tlačítko „Upravit kategorii“.
3. Načti z relace „Category“ všechny jména (kromě jména aktuálně editované kategorie) dosud použita v rodičovské kategorii. Nemá-li vybraná kategorie rodiče, pak načti všechny jména kořenových („root“) kategorií. Otevři dialog pro úpravu kategorie, předej načtená jména do tohoto dialogu, vyplň jméno a popis, dle hodnot z vybrané kategorie a umožni změnu jména a popisu kategorie. Dialog se potvrdí kliknutím na tlačítko „OK“, stornuje se tlačítkem „Storno“.
4. Klikne-li uživatel na tlačítko „OK“, pokračuj dalším bodem. Jinak uzavři dialog a ukonči funkci.
5. Ověř, zdali jméno kategorie je jedinečné, zdali je zadáno a zdali nezačíná mezerou, popis není nutno zadávat. Je-li některá z předešlých kontrol pozitivní, pak upozorni uživatele na daný problém a požaduj opravu zadání. Jinak uzavři dialog editace a pokračuj dále.
6. Uprav hodnoty atributů „cat\_name“ a „cat\_description“ dle hodnot zadanych v dialogu v záznamu v relaci „Category“ s „cat\_id“ = „cat\_id“ editované kategorie.
7. Proved' aktualizaci stromu kategorií a ukonči funkci

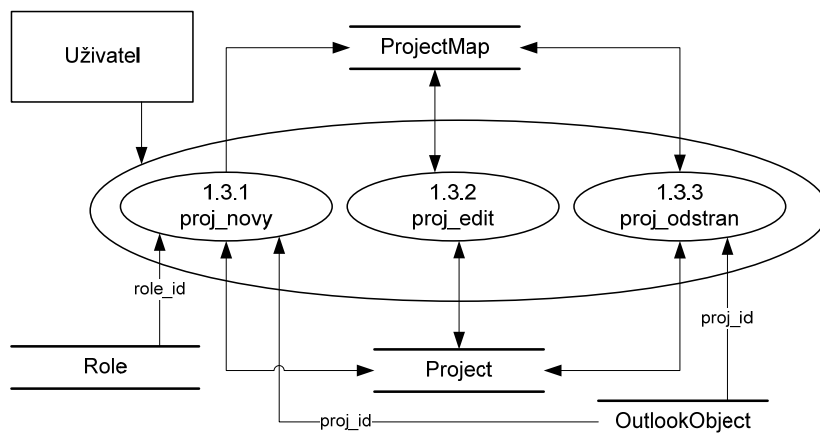
#### **4.1.6.2.4.3 Návrh algoritmu subsystému „1.2.3 kat\_odstran“**

1. První kroky jsou shodné s kroky 1 -5 popsány v předchozí kapitole (viz. 4.1.6.2.4.1 Návrh algoritmu subsystému „1.2.1 kat\_nova“).
2. Uživatel ze stromu vybere jednu ze zobrazených kategorií a klikne na tlačítko „Odstranit kategorii“.
3. Zobraz dotaz, zdali je si uživatel vědom toho, že daná kategorie bude odstraněna. Je-li odpověď kladná, pokračuj dále. Jinak ukonči funkci.
4. Ověř, prostřednictvím hodnoty atributu „cat\_id“ vybrané kategorie zdali se tato kategorie, kterou uživatel požaduje odstranit, nevyskytuje v relaci „OutlookObject“ ani není použita v relaci „CategoryMap“ jako hodnota atributu „cat\_parent\_id“.

Vyskytuje-li se alespoň v jedné z uvedených podmínek, oznam uživateli, že daná kategorie je využívána a tudíž nemůže být smazána. V tomto případě ukončí funkci, jinak pokračuj dále.

5. Odstraň z relace „Category“ záznam s hodnotou „cat\_id“ = „cat\_id“ aktuálně vybrané kategorie. Rovněž vyskytuje-li se v relaci „CategoryMap“ záznam s „cat\_id“ = „cat\_id“ vybrané kategorie odstraň i tento záznam. Odstranění záznamů z obou relací musí být opět ošetřeno v rámci transakce.
6. Proveď aktualizaci stromu kategorií a ukonči funkci

#### 4.1.6.2.5 DFD 3. úrovně – funkce subsystému „1.3 proj\_sprava“



Obr. 15: DFD 3. řádu (subsystémy náležející do 2. subsystému „proj\_sprava“)

##### 4.1.6.2.5.1 Návrh algoritmu subsystému „1.3.1 proj\_novy“

1. Uživatel otevře buďto panel GTD pro reporty nebo objekt aplikace MS Outlook, na který se vztahuje možnost zařazení do systému GTD (došlý email, připomínka v kalendáři a úkol).
2. Zobraz dle předchozího výběru uživatele příslušnou lištu nástrojů, která obsahuje mimo jiné tlačítko „Administrace projektů a kategorií“.
3. Uživatel kliknutím na toto tlačítko v liště nástrojů otevře dialogové okno pro správu projektů.

4. Zobraz dialogové okno, které bude obsahovat nahoře lištu tlačítek a v ní tlačítka pro správu projektů (tlačítka „Přidat projekt“, „Upravit projekt“ nebo „Odstranit projekt“).
5. Pod lištou nabídek se nachází komponenta pro vykreslení stromu. Zobraz zde stromovou strukturu projektů v závislosti na rolích, tak, aby role byly jako kořeny a na ně pak navazovaly příslušné projekty (dle návaznosti dané relací „ProjectMap“) – funkce pro konstrukci a zobrazení stromu bude blíže specifikována v subsystému „2.2 rep\_projekt“.
6. Uživatel vybere ze stromu buďto roli, nebo projekt, který bude sloužit jako rodičovský(á) pro aktuálně přidávaný projekt. Pak klikne na tlačítko „Přidat projekt“.
7. Pokud je jako rodič nově vznikajícího projektu vybrán také projekt, zkontroluj, zdali hodnota atributu „proj\_id“ rodičovského projektu není obsažena v relaci „OutlookObject“ v atributu „proj\_id“. V případě že ano, oznam uživateli, že daný projekt nemůže být použit jako rodič, jelikož obsahuje objekty MS Outlook, a ukonči vykonávání funkce.
8. Načti z relace „Project“ všechna jména dosud použitá v rodičovském projektu. Nemá-li vybraný projekt rodiče, pak načti všechna jména kořenových („root“) projektů. Rovněž získej hodnotu atributu „proj\_id“ a „role\_id“ z vybraného rodičovského projektu nebo role (v případě role zůstává hodnota atributu „proj\_id“ prázdná). Otevři dialog pro přidání projektů, předej dříve načtené hodnoty do tohoto dialogu a doplň hodnoty do combo boxů pro výběr priority (Nízká, Normální, Vysoká) a statusu projektu (K posouzení, K vypracování, Rozpracován, Delegován, Dokončen, Zrušen). Pak požaduj zadání jména (text box), popisu (text box), statusu (combo box), priority (combo box), koncového data dokončení (kalendář) a stavu rozpracovanosti (box na číslo 0-100). Dialog se potvrdí kliknutím na tlačítko „OK“, stornuje se tlačítkem „Storno“.
9. Klikne-li uživatel na tlačítko „OK“, pokračuj dalším bodem. Jinak uzavři dialog a ukonči funkci.
10. Ověř, zdali jméno projektu je jedinečné v daném kontextu, zdali je zadáno a zdali nezačíná mezerou, popis není nutno zadávat. Rovněž ověř správnost zadání stavu rozpracovanosti projektu. Hodnoty v comboboxech nebude možno zapisovat, půjdou pouze vybrat – prázdný výběr je tedy zakázán. Rovněž datum bude

kontrolován (použitou komponentou) je-li jeho formát správný. Je-li některá z předešlých kontrol pozitivní, pak upozorní uživatele na daný problém a požadují opravu zadání. Jinak uzavře dialog pro vytvoření nového projektu a pokračuje dále.

11. Vytvoř nový záznam v relaci „Project“ a zapiš hodnoty zadané v dialogu do atributů „proj\_name“, „proj\_description“, „proj\_priority“, „proj\_progress“, „proj\_status“ a „proj\_due\_date“, doplň také atribut „role\_id“ dle vybraného rodiče. Byl-li jako rodič vybrán projekt, pak získej hodnotu „proj\_id“ nově vytvořeného projektu a proved' zápis nového záznamu do relace „ProjectMap“, kde atribut „proj\_id“ je roven hodnotě atributu „proj\_id“ nově vytvořeného záznamu v relaci „Project“ a atribut „proj\_parent\_id“ = hodnotě atributu „proj\_id“ rodičovského projektu. Zápis obou záznamů do relací „Project“ a „ProjectMap“ je nutno provést v rámci transakce (jde o dva na sobě závislé záznamy).
12. Proveď aktualizaci stromu projektů a ukonči funkci.

#### **4.1.6.2.5.2 Návrh algoritmu subsystému „1.3.2 proj\_edit“**

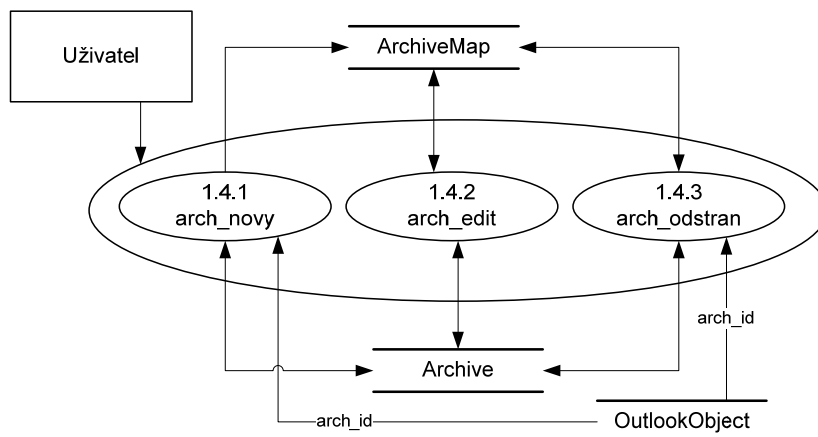
1. První kroky jsou shodné s kroky 1 -5 popsány v předchozí kapitole (viz. 4.1.6.2.5.1 Návrh algoritmu subsystému „1.3.1 proj\_novy“).
2. Uživatel vybere ze stromu projekt, který chce měnit a klikne na tlačítko „Upravit projekt“
3. Načti z relace „Project“ všechna jména dosud použitá v rodičovském projektu (kromě jména aktuálně editovaného projektu). Nemá-li vybraný projekt rodiče, pak načti všechna jména kořenových („root“) projektů (opět s výjimkou jména aktuálně editovaného projektu). Otevři dialog pro úpravu projektů, předej dříve načtené hodnoty do tohoto dialogu a doplň hodnoty do comboboxů pro výběr priority (Nízká, Normální, Vysoká) a statusu projektu (K posouzení, K vypracování, Rozpracován, Delegován, Dokončen, Zrušen). Je-li editovaný projekt koncovým projektem (nejsou na něm závislé další projekty), pak umožni editaci všech výše zmíněných atributů. Obsahuje-li pak aktuálně editovaný projekt další projekty, povol pouze změnu jména a popisu projektu (ostatní atributy zde nejsou žádané z důvodu jejich doplňování ze sub-projektů dle pravidel uvedených dále).
4. Uživatel potvrdí dialog kliknutím na tlačítko „OK“, stornuje se tlačítkem „Storno“.
5. Pokračuj dále, kliknul-li uživatel na „OK“, jinak uzavře dialog a ukonči funkci.

6. Ověř, zdali jméno projektu je jedinečné v daném kontextu, zdali je zadáno a zdali nezačíná mezerou, popis není nutno zadávat. Rovněž ověř správnost zadání stavu rozpracovanosti projektu. Hodnoty v comboboxech nebude možno zapisovat, půjdou pouze vybrat – prázdný výběr je tedy zakázán. Rovněž datum bude kontrolován (použitou komponentou) je-li jeho formát správný. Je-li některá z předešlých kontrol pozitivní, pak upozorni uživatele na daný problém a požaduj opravu zadání. Jinak uzavři dialog pro vytvoření nového projektu a pokračuj dále.
7. Uprav záznam v relaci „Project“ kde „proj\_id“ = „proj\_id“ aktuálně editovaného projektu, a změn hodnoty atributů „proj\_name“, „proj\_description“, „proj\_priority“, „proj\_progress“, „proj\_status“ a „proj\_due\_date“ dle hodnot zadaných v dialogu.
8. Proveď aktualizaci stromu projektů a ukonči funkci

#### **4.1.6.2.5.3 Návrh algoritmu subsystému „1.3.3 proj\_odstran“**

1. První kroky jsou shodné s kroky 1 -5 popsány v předchozí kapitole (viz. 4.1.6.2.5.1 Návrh algoritmu subsystému „1.3.1 proj\_novy“).
2. Uživatel ze stromu vybere jeden ze zobrazených projektů a klikne na tlačítko „Odstranit projekt“.
3. Zobraz dotaz, zdali je si uživatel vědom toho, že daný projekt bude odstraněn. Je-li odpověď kladná, pokračuj dále. Jinak ukonči funkci.
4. Ověř, prostřednictvím hodnoty atributu „proj\_id“ vybraného projektu zdali se tento projekt, který uživatel požaduje odstranit, nevyskytuje v relaci „OutlookObject“ ani není použit v relaci „ProjectMap“ jako hodnota atributu „proj\_parent\_id“. Vyskytuje-li se alespoň v jedné z uvedených podmínek, oznam uživateli, že daný projekt je využíván a tudíž nemůže být odstraněn. V tomto případě ukonči funkci, jinak pokračuj dále.
5. Odstraň z relace „Project“ záznam s hodnotou „proj\_id“ = „proj\_id“ aktuálně vybraného projektu. Rovněž vyskytuje-li se v relaci „ProjectMap“ záznam s „proj\_id“ = „proj\_id“ vybraného projektu odstraň i tento záznam. Odstranění záznamů z obou relací musí být opět ošetřeno v rámci transakce.
6. Proveď aktualizaci stromu projektů a ukonči funkci

#### 4.1.6.2.5.4 DFD 3. úrovně – funkce subsystému „1.4 arch\_sprava“



Obr. 16: DFD 3. řádu (subsystémy náležející do 2. subsystému „arch\_sprava“)

#### 4.1.6.2.5.5 Návrh algoritmu subsystému „1.4.1 arch\_novy“

1. Uživatel otevře buďto „task pane“ pro reporty nebo objekt aplikace MS Outlook, na který se vztahuje možnost zařazení do systému GTD (došlý email, připomínka v kalendáři a úkol).
2. Zobraz dle předchozího výběru uživatele příslušnou lištu nástrojů, která obsahuje mimo jiné tlačítko „Administrace archívů“.
3. Uživatel kliknutím na toto tlačítko v liště nástrojů otevře dialogové okno pro správu archívů.
4. Zobraz dialogové okno, které bude obsahovat nahoře lištu tlačítek a v ní tlačítka pro správu archívů (tlačítka „Přidat archív“, „Upravit archív“ nebo „Odstranit archív“).
5. Pod lištou nabídek se nachází komponenta pro vykreslení stromu. Zobraz zde stromovou strukturu archívů dle návaznosti dané relací „ArchiveMap“ – funkce pro konstrukci a zobrazení stromu bude blíže specifikována v subsystému „2.3 rep\_archiv“.
6. Uživatel vybere buďto existující archív a pak pokračuje příkazem „Přidat archív“ z nástrojové lišty, nebo vyžaduje vytvořit nový archív jako jeden z hlavních kořenových („root“) archívů a pak vybere z kontextového menu (zobrazí jej kliknutím pravým tlačítkem nad stromem archívů) příkaz „Přidat hlavní archív“.



7. Pokud je jako rodič nově vznikajícího archívu vybrán také archív, zkontroluj, zdali hodnota atributu „arch\_id“ rodičovského archívu není obsažena v relaci „OutlookObject“ v atributu „arch\_id“. V případě že ano, oznam uživateli, že daný archív nemůže být použit jako rodič, jelikož obsahuje objekty MS Outlook, a ukonči vykonávání funkce.
8. Načti z relace „Archive“ všechny jména dosud použitá v rodičovském archívu. Nemá-li vybraný archív rodiče, pak načti všechny jména kořenových („root“) archívů. Rovněž získej hodnotu atributu „arch\_id“ z vybraného rodičovského archívu (v případě, že archív bude kořenovým, pak hodnota rodičovského „arch\_id“ bude prázdná). Otevři dialog pro přidání archívu, předej dříve načtené hodnoty do tohoto dialogu a požaduj zadání jména a popisu archívu. Dialog se potvrdí kliknutím na tlačítko „OK“, stornuje se tlačítkem „Storno“.
9. Klikne-li uživatel na tlačítko „OK“, pokračuj dalším bodem. Jinak uzavři dialog a ukonči funkci.
10. Ověř, zdali jméno archívu je jedinečné v daném kontextu, zdali je zadáno a zdali nezačíná mezerou, popis není nutno zadávat. Je-li některá z předešlých kontrol pozitivní, pak upozorni uživatele na daný problém a požaduj opravu zadání. Jinak uzavři dialog pro vytvoření nového archívu a pokračuj dále.
11. Vytvoř nový záznam v relaci „Archive“ a doplň hodnoty atributů „arch\_name“ a „arch\_description“ dle údajů zadaných prostřednictvím dialogu. Byl-li jako rodič vybrán archív, pak získej hodnotu „arch\_id“ nově vytvořeného archívu, a proved' zápis nového záznamu do relace „ArchiveMap“, kde atribut „arch\_id“ je roven hodnotě atributu „arch\_id“ nově vytvořeného záznamu v relaci „Archive“ a atribut „arch\_parent\_id“ = hodnotě atributu „arch\_id“ rodičovského archívu. Zápis obou záznamů do relací „Archive“ a „ArchiveMap“ je samozřejmě nutno provést v rámci transakce.
12. Proved' aktualizaci stromu archívů a ukonči funkci

#### **4.1.6.2.5.6 Návrh algoritmu subsystému „1.4.1 arch\_edit“**

1. První kroky jsou shodné s kroky 1 -5 popsány v předchozí kapitole (viz. 4.1.6.2.5.5 Návrh algoritmu subsystému „1.4.1 arch\_novy“).
2. Uživatel ze stromu vybere jeden ze zobrazených archívů a klikne na tlačítko „Upravit archív“.
3. Načti z relace „Archive“ všechny jména (kromě jména aktuálně editovaného archívu) dosud použitá v rodičovském archívu. Nemá-li vybraný archív rodiče, pak načti všechny jména kořenových („root“) archívu. Otevři dialog pro úpravu archívu, předej načtená jména do tohoto dialogu, vyplň jméno a popis, dle hodnot z vybraného archívu a umožni změnu jména a popisu archívu. Dialog se potvrdí kliknutím na tlačítko „OK“, stornuje se tlačítkem „Storno“.
4. Klikne-li uživatel na tlačítko „OK“, pokračuj dalším bodem. Jinak uzavři dialog a ukonči funkci.
5. Ověř, zdali jméno archívu je jedinečné, zdali je zadáno a zdali nezačíná mezerou, popis není nutno zadávat. Je-li některá z předešlých kontrol pozitivní, pak upozorni uživatele na daný problém a požaduj opravu zadání. Jinak uzavři dialog editace a pokračuj dále.
6. Uprav hodnoty atributů „arch\_name“ a „arch\_description“ (dle hodnot zadaných v dialogu) v záznamu v relaci „Category“ s „arch\_id“ = „arch\_id“ editovaného archívu.
7. Proveď aktualizaci stromu archívů a ukonči funkci

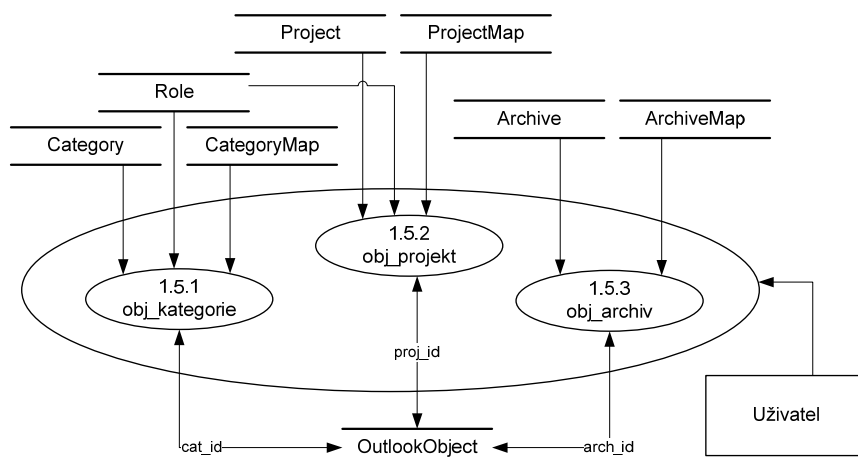
#### **4.1.6.2.5.7 Návrh algoritmu subsystému „1.4.2 arch\_odstran“**

1. První kroky jsou shodné s kroky 1 -5 popsány v předchozí kapitole (viz. 4.1.6.2.5.5 Návrh algoritmu subsystému „1.4.1 arch\_novy“).
2. Uživatel ze stromu vybere jeden ze zobrazených archívů a klikne na tlačítko „Odstranit archív“.
3. Zobraz dotaz, zdali je si uživatel vědom toho, že daný archív bude odstraněn. Je-li odpověď kladná, pokračuj dále. Jinak ukonči funkci.
4. Ověř, prostřednictvím hodnoty atributu „arch\_id“ vybraného archívu zdali se tento archív, který uživatel požaduje odstranit, nevyskytuje v relaci „OutlookObject“ ani není použit v relaci „ArchiveMap“ jako hodnota atributu „arch\_parent\_id“.

Vyskytuje-li se alespoň v jedné z uvedených podmínek, oznam uživateli, že daný archiv je využívána a tudíž nemůže být odstraněn. V tomto případě ukončí funkci, jinak pokračuj dále.

5. Odstraň z relace „Archive“ záznam s hodnotou „arch\_id“ = „arch\_id“ aktuálně vybraného archívu. Rovněž vyskytuje-li se v relaci „ArchiveMap“ záznam s „arch\_id“ = „arch\_id“ vybraného archívu odstraň i tento záznam. Odstranění záznamů z obou relací musí být opět ošetřeno v rámci transakce.
6. Proveď aktualizaci stromu kategorií a ukonči funkci

#### 4.1.6.2.6 DFD 3. úrovně – funkce subsystému „1.5 obj\_sprava“



Obr. 17: DFD 3. řádu (subsystémy náležející do 2. subsystému „obj\_sprava“)

##### 4.1.6.2.6.1 Návrh algoritmu subsystému „1.5.1 obj\_kategorie“

1. Uživatel otevře objekt MS Outlook, který je doplňkem podporován (doručená pošta, připomínka v kalendáři, úkol). V záložkách pro volbu nástrojové lišty zvolí záložku GTD
2. Zobraz nástrojovou lištu GTD:



Obr. 18: Návrh lišty nástrojů GTD

3. Z relace „Role“ vyber posledních X rolí, seřaď je podle hodnoty atributu „role\_last\_used“, a vygeneruj získané role jako položky menu „Zařadit do rolí“. Pokud není role vybrána, deaktivuj menu „Zařadit do kategorií“ a „Zařadit do projektu“
4. Uživatel klikne na menu „Zařadit do rolí“
5. Otevři menu a nabídni uživateli výběr rolí:



**Obr. 19: Menu "Zařadit do rolí" z lišty nástrojů GTD**

6. Uživatel vybere buďto přímo roli, nebo klikne na položku „Všechny role a kategorie“
7. Uživatel vybere přímo jednu z nabízených rolí, pak pokračuj dále. Vybere-li uživatel příkaz „Všechny role a kategorie“, pokračuj bodem 23.
8. Ulož do interní proměnné „mRoleId“ hodnotu atributu „role\_id“ aktuálně vybrané role.
9. Aktualizuj v relaci „Role“ hodnotu atributu „role\_last\_used“ na hodnotu aktuálního data v záznamu, kde hodnota atributu „role\_id“ = „mRoleId“.
10. Z relace „Category“ a „CategoryMap“ načti všechny kategorie (a jejich vzájemné mapování v rámci hierarchického stromu kategorií), jejichž hodnota atributu „role\_id“ = „role\_id“ vybrané role. Seřaď načtené kategorie dle hodnoty atributu „cat\_last\_used“ tak, aby posledně použitá kategorie byla na prvním místě.
11. Vygeneruj pro každou z kategorií plné jméno (tzn. jméno kategorie, včetně všech nadřazených rodičovských kategorií), uchovej toto jméno v instanci představující každou kategorii (použij např. vlastnost „FullName“), poté vyber posledních N<sup>4</sup> kategorií, které nejsou dále používány jako rodičovské (hodnota jejich atributu „cat\_id“ se nevyskytuje v hodnotě atributu „cat\_parent\_id“ v relaci „CategoryMap“) a vygeneruj z nich položky pro menu „Zařadit do kategorií“. Je-li

---

<sup>4</sup> Udává počet kategorií, které se mají v menu zobrazit. Počet bude dán konfiguračním souborem

počet takto vygenerovaných položek menu „Zařadit do kategorií“ > 0, pak povol menu „Zařadit do kategorií“, v opačném případě toto menu zakaž.

12. Z relace „Project“ a „ProjectMap“ načti všechny projekty (a jejich vzájemné mapování v rámci hierarchického stromu projektů), jejichž hodnota atributu „role\_id“ = „role\_id“ vybrané role. Seřaď načtené projekty dle hodnoty atributu „proj\_last\_used“ tak, aby posledně použitý projekt byl na prvním místě.
13. Vygeneruj pro každý z načtených projektů plné jméno (tzn. jméno projektu, včetně všech nadřazených rodičovských projektů), uchovej toto jméno v instanci představující každý projekt (již zmiňována vlastnost „FullName“), poté vyber posledních  $N^4$  projektů, které nejsou dále používány jako rodičovské (hodnota jejich atributu „proj\_id“ se nevyskytuje v hodnotě atributu „proj\_parent\_id“ v relaci „ProjectMap“) a vygeneruj z nich položky pro menu „Zařadit do projektů“. Je-li počet takto vygenerovaných položek menu „Zařadit do projektů“ > 0, pak povol menu „Zařadit do projektů“, v opačném případě toto menu zakaž.
14. Uživatel klikne na menu „Zařadit do kategorií“.
15. Otevři menu a nabídni uživateli výběr kategorií:



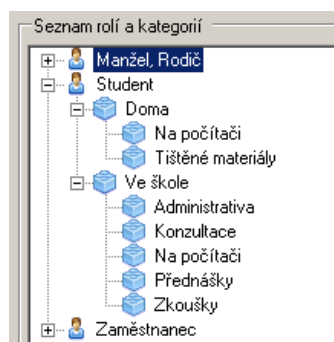
**Obr. 20: Menu "Zařadit do kategorií" z lišty nástrojů GTD**

16. Vybere-li uživatel jednu z nabízených kategorií, pak pokračuj dále. Pokud uživatel vybere příkaz „Všechny role a kategorie“ pokračuj bodem 23.
17. Ulož do vnitřních proměnných „mRoleId“ a „mCategoryId“ hodnoty atributů „role\_id“ a „cat\_id“ získané z vybrané kategorie.
18. V relaci „Category“ aktualizuj hodnotu atributu „cat\_last\_used“ na hodnotu aktuálního data v záznamu, kde hodnota atributu „cat\_id“ = „mCategoryId“.
19. Zobraz název vybrané kategorie ve skupině „Zařazení“:



**Obr. 21: Skupina "Zařazení" z lišty nástrojů GTD**

20. Zakaž možnost zařadit daný objekt aplikace MS Outlook do archívu znepřístupněním menu „Zařadit do archívu“
21. Jeli vybraná kategorie, pak aktivuj možnost zrušit toto zařazení pomocí tlačítka „X“ vedle názvu vybrané kategorie - v případě že jej uživatel použije, anuluj veškerá nastavení provedená při výběru kategorie, tj. anuluj hodnoty „mRoleId“ a „mCategoryId“ a u položky Kategorie ve skupině „Zařazení“ uveď hodnotu „Nespecifikovaná“. Opětovně také povol menu „Zařadit do archívu“.
22. Pokračuj bodem 33.
23. Uživatel použil příkaz „Všechny role a kategorie“.
24. Otevři dialogové okno se stromovým zobrazením kategorií závislých na rolích a s tlačítky „Ok“ a „Storno“:



**Obr. 22: Výřez z dialogu pro výběr kategorií**

25. Zakaž tlačítko „Ok“ v případě, že je ve stromu vybrána role nebo kategorie, která je rodičovská a obsahuje další kategorie (potomky).
26. Uživatel může výběr kategorie ukončit kliknutím na tlačítko „Storno“, pak zavři aktuální dialog a ukonči funkci, jinak pokračuj dále.
27. Uživatel vybere koncovou kategorií
28. Povol tlačítko „Ok“
29. Uživatel potvrdí výběr kategorie tlačítkem „Ok“
30. Zavři aktuální dialog a ulož hodnoty atributů „role\_id“ a „cat\_id“ vybrané kategorie (entita kategorie obsahuje i atribut „role\_id“, tudíž není nutno procházet strom ke

kořenu pro zjištění vybrané role) do vnitřních proměnných „mRoleId“ a „mCategoryId“.

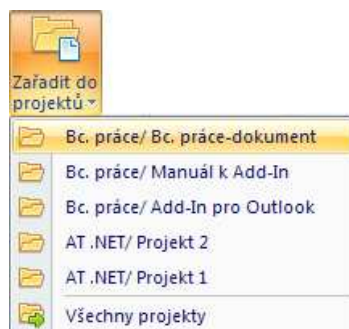
31. Aktualizuj v relaci „Role“ hodnotu atributu „role\_last\_used“ na hodnotu aktuálního data v záznamu, kde hodnota atributu „role\_id“ = „mRoleId“.
32. Postupuj shodně s kroky 18 - 21 aktuální kapitoly.
33. Uživatel může dále pokračovat přiřazením projektu k aktuálně zpracovávanému objektu aplikace MS Outlook (viz 4.1.6.2.6.2 Návrh algoritmu subsystému „1.5.2 obj\_projekt“) nebo může proceduru přiřazení kategorie ukončit buďto schválením nebo ignorováním provedených změn
34. Uzavře-li v tomto okamžiku uživatel okno aktuálně zpracovávaného objektu MS Outlook, pak ignoruj veškeré změny, které uživatel provedl, a ukonči funkci. Jinak pokračuj dále.
35. Uživatel klikne na tlačítko „Akceptovat zařazení“ ve skupině „Zařazení“. Získej z prostředí MS Outlook ID aktuálně zpracovávaného Outlook objektu a ulož do interní proměnné „mEntryId“.
36. Ověř, zdali v relaci „OutlookObject“ existuje záznam s hodnotou „obj\_entry\_id“ = „mEntryId“. Existuje-li, pak uprav v tomto záznamu hodnoty atributů „cat\_id“, „proj\_id“, „arch\_id“ dle hodnot interních proměnných „mCategoryId“, „mProjectId“<sup>5</sup> a „mArchiveId“<sup>5</sup>
37. Neexistuje-li, pak vlož nový záznam s hodnotou atributů „obj\_entry\_id“ = „mEntryId“, „cat\_id“ = „mCategoryId“, „proj\_id“ = „mProjectId“ a „arch\_id“ = „mArchiveId“ do relace „OutlookObject“
38. Ukonči funkci zařazování, uživatel může uzavřít aktuálně zpracovávaný objekt aplikace MS Outlook, zařazení objektu bylo provedeno

---

<sup>5</sup> Nastavení obsahu interní proměnné „mProjectId“ bude definován dále (viz. 4.1.6.2.6.2 Návrh algoritmu subsystému „1.5.2 obj\_projekt“) a taktéž „mArchiveId“ bude definován dále (viz. 4.1.6.2.6.3 Návrh algoritmu subsystému „1.5.3 obj\_archív“) - nicméně v rámci zařazení objektu MS Outlook se funkce zařazení do kategorií, zařazení do projektů a zařazení do archívu těsně prolínají a koexistují v daném kontextu aktuálně editovaného objektu MS Outlook.

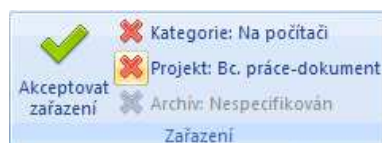
#### 4.1.6.2.6.2 Návrh algoritmu subsystému „1.5.2 obj\_projekt“

1. Zařazení objektů aplikace Outlook do projektů je rovněž závislé na výběru role. Tudíž body 1 - 13 předchozí kapitoly (viz. 4.1.6.2.6.1 Návrh algoritmu subsystému „1.5.1 obj\_kategorie“) jsou nutným předpokladem pro tento algoritmus.
2. Uživatel klikne na menu „Zařadit do projektů“
3. Zobraz menu a nabídni uživateli výběr projektu:



**Obr. 23: Menu "Zařadit do projektů" z lišty nástrojů GTD**

4. Vybere-li si uživatel jeden z nabízených projektů, pokračuj dále. Vybere-li si uživatel příkaz „Všechny projekty“, pak pokračuj bodem 11.
5. Ulož do vnitřních proměnných „mRoleId“ a „mProjectId“ hodnoty atributů „role\_id“ a „proj\_id“ získané z vybraného projektu
6. V relaci „Project“ aktualizuj hodnotu atributu „proj\_last\_used“ na hodnotu aktuálního data v záznamu, kde hodnota atributu „proj\_id“ = „mProjectId“.
7. Zobraz název vybraného projektu ve skupině „Zařazení“:

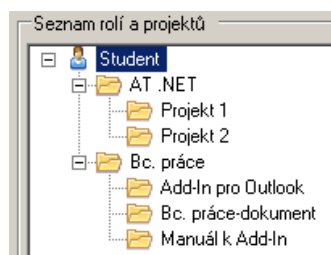


**Obr. 24: Skupina "Zařazení" z lišty nástrojů GTD - přidán projekt**

8. Zakaž možnost zařadit daný objekt aplikace MS Outlook do archívu znepřístupněním menu „Zařadit do archívu“.
9. Je-li vybrán projekt, pak aktivuj možnost zrušit toto zařazení pomocí tlačítka „X“ vedle názvu vybraného projektu - v případě že jej uživatel použije, anuluj veškerá nastavení provedená při výběru projektu, tj. anuluj hodnotu „mProjectId“ a u položky „Projekt“ ve skupině „Zařazení“ uveď hodnotu „Nespecifikován“. V případě, že není specifikována ani kategorie, opětovně povol menu „Zařadit do archívu“.



10. Pokračuj bodem 20.
11. Uživatel použil příkaz „Všechny projekty“.
12. Otevři dialogové okno se stromovým zobrazením projektů závislých na aktuálně vybrané roli (viz bod 1 aktuální kapitoly) a s tlačítky „Ok“ a „Storno“:



**Obr. 25: Výřez z dialogu pro výběr projektů**

13. Zakaž tlačítko „Ok“ v případě, že je ve stromu vybrána role nebo projekt, který je rodičovský a obsahuje další sub-projekty (potomky).
14. Uživatel může výběr projektů ukončit kliknutím na tlačítko „Storno“, pak zavře aktuální dialog a ukončí funkci, jinak pokračuj dále.
15. Uživatel vybere koncový projekt
16. Povol tlačítko „Ok“
17. Uživatel potvrdí výběr kategorie tlačítkem „Ok“
18. Zavře aktuální dialog a uloží hodnotu atributu „proj\_id“ vybraného projektu.
19. Postupuj shodně s kroky 6 - 9 aktuální kapitoly.
20. Uživatel může proceduru přiřazení projektů ukončit buďto schválením nebo ignorováním provedených změn
21. Uzavře-li v tomto okamžiku uživatel okno aktuálně zpracovávaného objektu MS Outlook, pak ignoruj veškeré změny, které uživatel provedl, a ukončí funkci. Jinak pokračuj dále.
22. Uživatel klikne na tlačítko „Akceptovat zařazení“ ve skupině „Zařazení“.
23. Získej z prostředí MS Outlook ID aktuálně zpracovávaného Outlook objektu a ulož do interní proměnné „mEntryId“.
24. Ověř, zdali v relaci „OutlookObject“ existuje záznam s hodnotou „obj\_entry\_id“ = „mEntryId“. Existuje-li, pak uprav v tomto záznamu hodnoty atributů „cat\_id“,

„proj\_id“, „arch\_id“ dle hodnot interních proměnných „mCategoryId“, „mProjectId“ a „mArchiveId“<sup>6</sup>.

25. Neexistuje-li, pak vlož nový záznam s hodnotou atributů „obj\_entry\_id“ = „mEntryId“, „cat\_id“ = „mCategoryId“, „proj\_id“ = „mProjectId“ a „arch\_id“ = „mArchiveId“ do relace „OutlookObject“
26. Ukonči funkci zařazování, uživatel může uzavřít aktuálně zpracovávaný objekt aplikace MS Outlook, zařazení objektu bylo provedeno

#### 4.1.6.2.6.3 Návrh algoritmu subsystému „1.5.3 obj\_archív“

1. Z relace „Archive“ a „ArchiveMap“ načti všechny archívy (a jejich vzájemné mapování v rámci hierarchického stromu archívů). Seřaď načtené projekty dle hodnoty atributu „arch\_last\_used“ tak, aby posledně použitý archív byl na prvním místě.
2. Vygeneruj pro každý z načtených archívu plné jméno (tzn. jméno archívu, včetně všech nadřazených rodičovských archívu), uchovej toto jméno v instanci představující každý archív (již zmiňována vlastnost „FullName“), poté vyber posledních  $N^4$  archívů, které nejsou dále používány jako rodičovské (hodnota jejich atributu „arch\_id“ se nevyskytuje v hodnotě atributu „arch\_parent\_id“ v relaci „ArchiveMap“) a vygeneruj z nich položky pro menu „Zařadit do archívů“. Je-li počet takto vygenerovaných položek menu „Zařadit do archívu“ > 0, pak povol menu „Zařadit do archívu“, v opačném případě toto menu zakaž.
3. Uživatel klikne na menu „Zařadit do archívu“.
4. Otevři menu a nabídni uživateli výběr archívů:

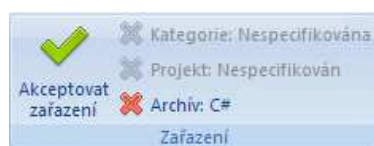
---

<sup>6</sup> Nastavení obsahu interní proměnné „mArchiveId“ bude definován dále (viz. 4.1.6.2.6.3 Návrh algoritmu subsystému „1.5.3 obj\_archív“)



**Obr. 26: Menu "Zařadit do archivu" z lišty nástrojů GTD**

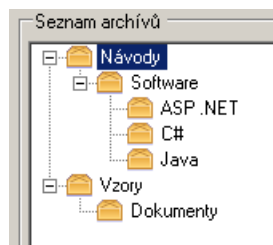
5. Vybere-li si uživatel jeden z nabízených archivů, pokračuj dále. Vybere-li si uživatel příkaz „Všechny projekty“, pak pokračuj bodem 12.
6. Ulož do vnitřní proměnné „mArchiveId“ hodnoty atributu „arch\_id“ získanou z vybraného archivu.
7. V relaci „Archive“ aktualizuj hodnotu atributu „arch\_last\_used“ na hodnotu aktuálního data v záznamu, kde hodnota atributu „arch\_id“ = „mArchiveId“.
8. Zobraz název vybraného archivu ve skupině „Zařazení“:



**Obr. 27: Skupina "Zařazení" z lišty nástrojů GTD – přiřazen archiv**

9. Zakaž možnost přiřadit danému objektu aplikace MS Outlook kategorií nebo projekt znepřístupněním menu „Zařadit do kategorií“ a „Zařadit do projektů“.
10. Je-li vybrán archiv, pak aktivuj možnost zrušit toto zařazení pomocí tlačítka „X“ vedle názvu vybraného archivu - v případě že jej uživatel použije, anuluj veškerá nastavení provedená při výběru archivu, tj. anuluj hodnotu „mArchiveId“ a u položky „Archív“ ve skupině „Zařazení“ uveď hodnotu „Nespecifikován“. Rovněž umožni opětovné zadání kategorie a projektů povolením menu „Zařadit do kategorií“ a „Zařadit do projektů“
11. Pokračuj bodem 21.
12. Uživatel použil příkaz „Všechny archivy“.

13. Otevří dialogové okno se stromovým zobrazením archívů a s tlačítky „Ok“ a „Storno“:

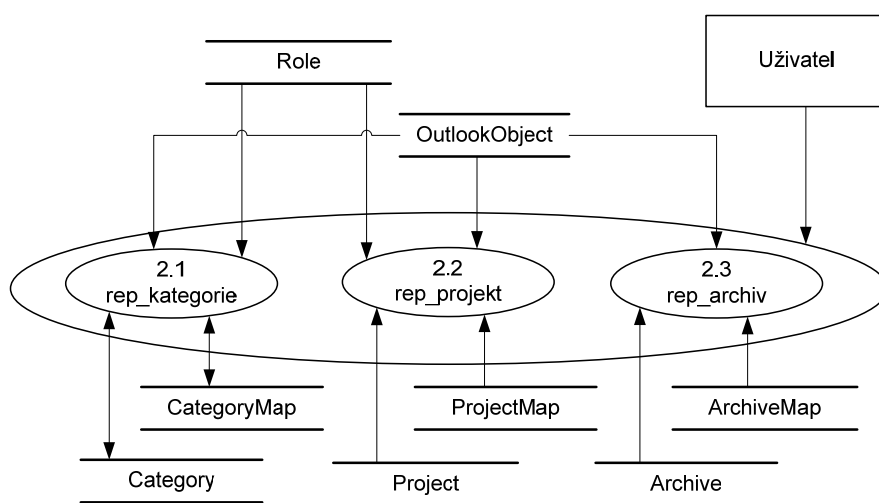


**Obr. 28: Výřez z dialogu pro výběr archívů**

14. Zakaž tlačítko „Ok“ v případě, že je ve stromu vybrán archív, který je rodičovský a obsahuje další sub-archívy (potomky).
15. Uživatel může výběr archívu ukončit kliknutím na tlačítko „Storno“, pak zavři aktuální dialog a ukonči funkci, jinak pokračuj dále.
16. Uživatel vybere koncový archív.
17. Povol tlačítko „Ok“.
18. Uživatel potvrdí výběr kategorie tlačítkem „Ok“.
19. Zavři aktuální dialog.
20. Postupuj shodně s kroky 6 - 10 aktuální kapitoly.
21. Uživatel může proceduru přiřazení archívů ukončit buďto schválením nebo ignorováním provedených změn
22. Uzavře-li v tomto okamžiku uživatel okno aktuálně zpracovávaného objektu MS Outlook, pak ignoruj veškeré změny, které uživatel provedl, a ukonči funkci. Jinak pokračuj dále.
23. Uživatel klikne na tlačítko „Akceptovat zařazení“ ve skupině „Zařazení“.
24. Získej z prostředí MS Outlook ID aktuálně zpracovávaného Outlook objektu a ulož do interní proměnné „mEntryId“.
25. Ověř, zdali v relaci „OutlookObject“ existuje záznam s hodnotou „obj\_entry\_id“ = „mEntryId“. Existuje-li, pak uprav v tomto záznamu hodnoty atributů „cat\_id“, „proj\_id“, „arch\_id“ dle hodnot interních proměnných „mCategoryId“, „mProjectId“ a „mArchiveId“.
26. Neexistuje-li, pak vlož nový záznam s hodnotou atributů „obj\_entry\_id“ = „mEntryId“, „cat\_id“ = „mCategoryId“, „proj\_id“ = „mProjectId“ a „arch\_id“ = „mArchiveId“ do relace „OutlookObject“

27. Ukonči funkci zařazování, uživatel může uzavřít aktuálně zpracováváný objekt aplikace MS Outlook, zařazení objektu bylo provedeno

#### 4.1.6.2.7 DFD 2. úrovně – funkce subsystému „2 report“



Obr. 29: DFD 2. řádu (subsystémy náležející do 1. subsystému „report“)

##### 4.1.6.2.7.1 Návrh algoritmu subsystému „2.1 rep\_kategorie“

1. Doplně do prvního comboboxu v panelu pro reportování aktuální seznam rolí.
2. Doplně do druhého comboboxu v panelu pro reportování volby „Přehled kategorií“, „Přehled projektů“ a „Přehled archívů“.
3. Zobraz panel pro reportování:



Obr. 30: Lišta nástrojů panelu pro reportování

4. Uživatel vybere roli, v jaké chce aktuálně vystupovat vůči systému.
5. Ulož hodnotu atributu „role\_id“ do interní proměnné „mRoleId“.
6. Uživatel z přehledů vybere „Přehled kategorií“.
7. Uživatel potvrdí výběr.
8. Načti všechny kategorie z relace „Category“ a jejich vzájemné vazby z relace „CategoryMap“.

9. Načti všechny odkazy na Outlook objekty z relace „OutlookObject“, kde hodnota atributu „cat\_id“ je v poli všech hodnot atributu „cat\_id“ z relace „Category“, kde hodnota atributu „role\_id“ = „mRoleId“. Ulož tyto objekty do interního seznamu „mObjectList“
10. Zkonstruuj strom kategorií dle vzájemných vazeb dříve načtených z relace „CategoryMap“ následujícím způsobem:
  - a. Proveď sloučení tabulek „Category“ a „CategoryMap“ tak, aby všechny takto získané entity obsahovaly také atribut „cat\_parent\_id“, a to i v případě, že se nenajde, v případě entity z relace „Category“, žádná souvislost s entitami relace „CategoryMap“. I v tomto případě musí být daná entita obsažená v seznamu takto získaných entit a hodnota atributu „cat\_parent\_id“ této entity bude nastavena na zápornou hodnotu.
  - b. Ulož entity získané předchozím sloučením obou relací do interního seznamu „mCategoryList“.
  - c. Procházej všechny záznamy v seznamu „mCategoryList“ a:
    - i. Je-li hodnota atributu „cat\_parent\_id“ aktuálního záznamu záporná, daná kategorie nemá rodiče a je tedy kořenovou kategorií. Přidej ji tedy jako hlavní větev do konstruovaného stromu.
    - ii. Je-li hodnota atributu „cat\_parent\_id“ aktuálního záznamu větší než nula, pak je daná kategorie sub-kategorií. Vyhledej v aktuálním stromu jejího rodiče, jehož hodnota atribut „cat\_id“ = hodnotě atributu „cat\_parent\_id“ aktuálního záznamu a napoj danou kategorií na příslušnou rodičovskou kategorií
11. Projdi všechny objekty ze seznamu „mObjectsList“, najdi dle atributu „obj\_entry\_id“ skutečné objekty aplikace MS Outlook, získej z nich specifické informace jako stav, datum dokončení, prioritu, a proveď jejich napojení na odpovídající kategorie dle hodnoty atributu „cat\_id“ ve stromu kategorií.
12. Zobraz sestavený strom v panelu pro reportování.

#### **4.1.6.2.7.2 Návrh algoritmu subsystému „2.2 rep\_projekt“**

1. První kroky jsou shodné s kroky 1 - 5, kapitoly 4.1.6.2.7.1 Návrh algoritmu subsystému „2.1 rep\_kategorie“
2. Uživatel z přehledů vybere „Přehled projektů“.

3. Uživatel potvrdí výběr.
4. Načti všechny projekty z relace „Project“ a jejich vzájemné vazby z relace „ProjectMap“.
5. Načti všechny odkazy na Outlook objekty z relace „OutlookObject“, kde hodnota atributu „proj\_id“ je v poli všech hodnot atributu „proj\_id“ z relace „Project“, kde hodnota atributu „role\_id“ = „mRoleId“. Ulož tyto objekty do interního seznamu „mObjectList“
6. Zkonstruuuj strom projektů dle vzájemných vazeb dříve načtených z relace „ProjectMap“ následujícím způsobem:
  - a. Proveď sloučení relací „Project“ a „ProjectMap“ tak, aby všechny takto získané entity obsahovaly také atribut „proj\_parent\_id“, a to i v případě, že se nenajde u některé z entit relace „Project“ žádná souvislost s entitami relace „ProjectMap“. I v tomto případě musí být daná entita obsažená v seznamu takto získaných entit a hodnota atributu „proj\_parent\_id“ této entity bude nastavena na zápornou hodnotu.
  - b. Ulož entity získané předchozím sloučením obou relací do interního seznamu „mProjectList“.
  - c. Procházej všechny záznamy v seznamu „mProjectList“ a:
    - i. Je-li hodnota atributu „proj\_parent\_id“ aktuálního záznamu záporná, daný projekt nemá rodiče a je tedy kořenovým projektem. Přidej jej tedy jako hlavní větev do konstruovaného stromu.
    - ii. Je-li hodnota atributu „proj\_parent\_id“ aktuálního záznamu větší než nula, pak je daný projekt sub-projektem. Vyhledej v aktuálním stromu jeho rodiče, jehož hodnota atributu „proj\_id“ = hodnotě atributu „proj\_parent\_id“ aktuálního záznamu a napoj daný projekt na příslušný rodičovský projekt.
7. Projdi všechny objekty ze seznamu „mObjectsList“, najdi dle atributu „obj\_entry\_id“ skutečné objekty aplikace MS Outlook, získej z nich specifické informace jako stav, datum dokončení, prioritu, a proveď jejich napojení na odpovídající projekty dle hodnoty atributu „proj\_id“ ve stromu projektů.
8. Zobraz sestavený strom v panelu pro reportování.

#### 4.1.6.2.7.3 Návrh algoritmu subsystému „2.1 rep\_archive“

1. První kroky jsou shodné s kroky 1 - 3, kapitoly 4.1.6.2.7.1 Návrh algoritmu subsystému „2.1 rep\_kategorie“. Jak již bylo zmíněno dříve, archívy nejsou závislé na roli, tudíž výběr role zde není důležitý.
2. Uživatel z přehledů vybere „Přehled archívů“.
3. Uživatel potvrdí výběr.
4. Načti všechny archívy z relace „Archive“ a jejich vzájemné vazby z relace „ArchiveMap“.
5. Načti všechny odkazy na Outlook objekty z relace „OutlookObject“, kde hodnota atributu „arch\_id“ není prázdná. Ulož tyto objekty do interního seznamu „mObjectList“
6. Zkonstruuuj strom archívů dle vzájemných vazeb dříve načtených z relace „ProjectMap“ následujícím způsobem:
  - a. Proveď sloučení relací „Archive“ a „ArchiveMap“ tak, aby všechny takto získané entity obsahovaly také atribut „arch\_parent\_id“, a to i v případě, že se nenajde u některé z entit relace „Archive“ žádná souvislost s entitami relace „ArchiveMap“. I v tomto případě musí být daná entita obsažená v seznamu takto získaných entit a hodnota atributu „arch\_parent\_id“ této entity bude nastavena na zápornou hodnotu.
  - b. Ulož entity získané předchozím sloučením obou relací do interního seznamu „mArchiveList“.
  - c. Procházej všechny záznamy v seznamu „mArchiveList“ a:
    - i. Je-li hodnota atributu „arch\_parent\_id“ aktuálního záznamu záporná, daný archív nemá rodiče a je tedy kořenovým archívem. Přidej jej tedy jako hlavní větev do konstruovaného stromu.
    - ii. Je-li hodnota atributu „arch\_parent\_id“ aktuálního záznamu větší než nula, pak je daný archív sub-archívem. Vyhledej v aktuálním stromu jeho rodiče, jehož hodnota atributu „arch\_id“ = hodnotě atributu „arch\_parent\_id“ aktuálního záznamu a napoj daný archív na příslušný rodičovský archív.
7. Projdi všechny objekty ze seznamu „mObjectList“, najdi dle atributu „obj\_entry\_id“ skutečné objekty aplikace MS Outlook, získej z nich specifické



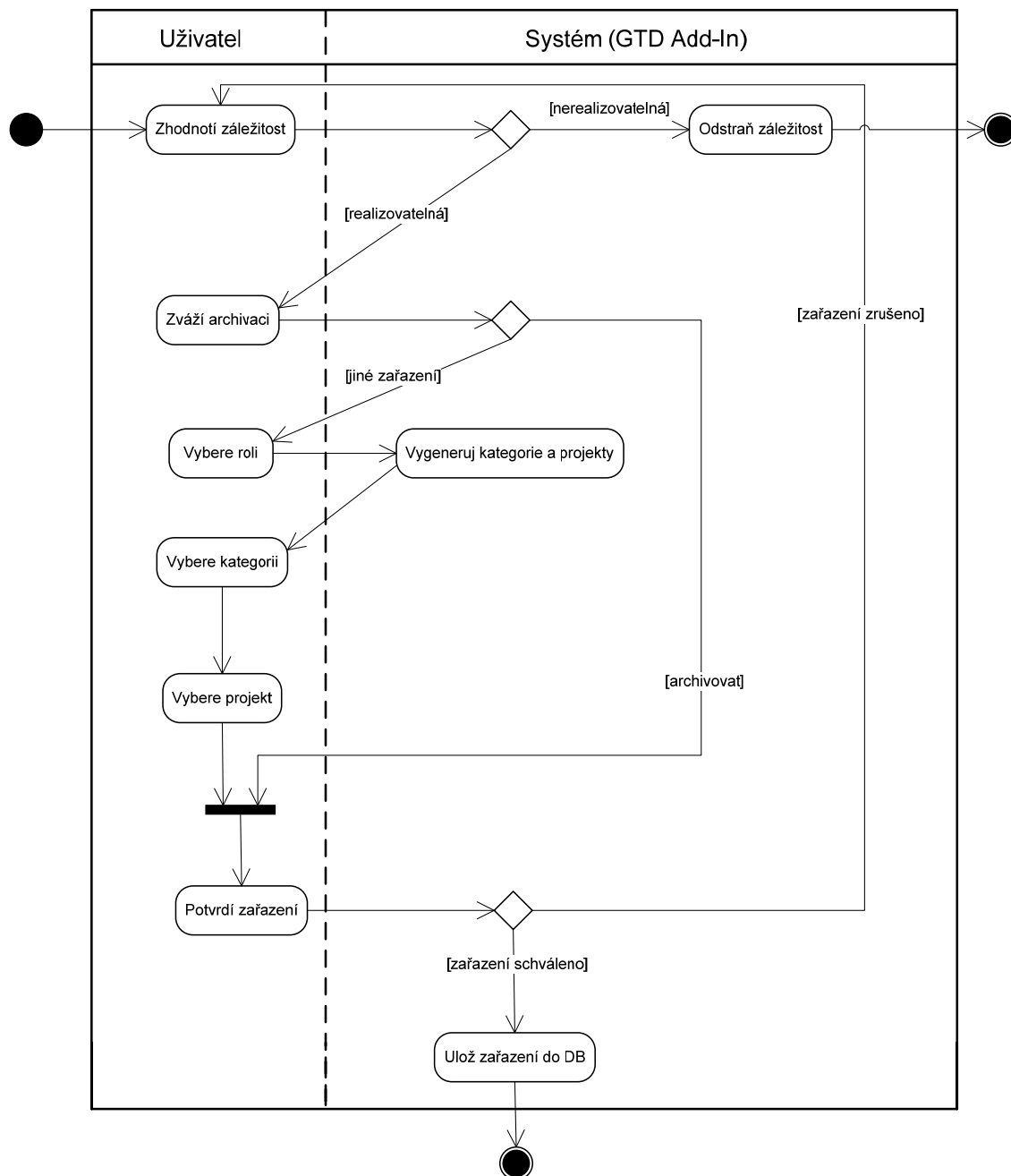
informace jako stav, datum dokončení, prioritu, a proved' jejich napojení na odpovídající projekty dle hodnoty atributu „proj\_id“ ve stromu projektů.

8. Zobraz sestavený strom v panelu pro reportování.

### 4.1.6.3 Diagramy aktivit

#### 4.1.6.4 Diagram aktivit subsystému „1.5 objekt\_sprava“

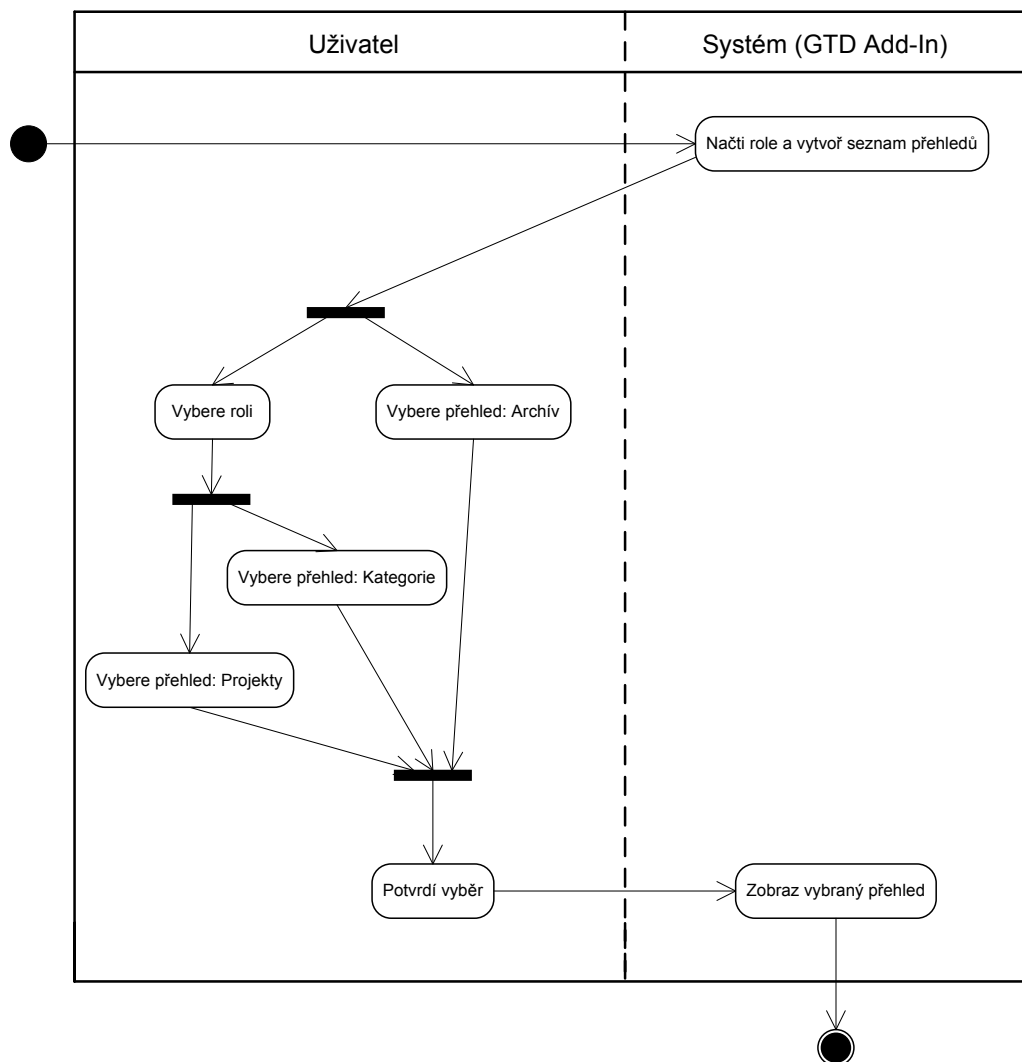
Tento diagram znázorňuje aktivity při zařazování objektu aplikace MS Outlook do GTD kategorií, projektů nebo archívů.



Obr. 31: Diagram aktivit subsystému „1.5 objekt\_sprava“

#### 4.1.6.5 Diagram aktivit subsystému „2 report“

Diagram znázorňuje úkony spojené s vytvoření požadované sestavy



Obr. 32: Diagram aktivit subsystému "2 report"

#### **4.1.7 Analýza datových toků**

Při návrhu algoritmů nebylo nutné definovat žádný další datový tok. Datové toky uvedené v DFD obsahují buďto celé entity (není-li datový tok uveden), anebo atribut dané entity, který je v příslušném datovém toku uveden.

#### **4.1.8 Indexová analýza**

Pro použití indexů umožňujících rychlejší vyhledávání požadovaných entit v rámci relací je nutno zvážit počet záznamů, které se budou v dané relaci nacházet a jak často bude dle atributu, který požadujeme indexovat, prováděno vyhledávání.

Režie na indexaci malých tabulek bývají větší než samotný přínos indexace, proto se pro relace s malým počtem záznamů nehodí.

Vyhledávání v relacích, jež jsou předmětem této analýzy, je prováděno v naprosté většině dle ID záznamu. Tyto identifikátory záznamů proto budeme indexovat všechny. Indexovány budou i identifikátory představující v dané relaci cizí klíče (tj. pomocí nich je prováděno spojování relací, vazba mezi relacemi apod.)

#### **4.1.9 Transakční analýza**

Transakčně je nutno ošetřit provádění těch databázových operací, při kterých se provádí vícero na sobě závislých zápisů do databáze. Transakce tímto pomáhá minimalizovat riziko vzniku nekonzistence dat v databázi. Pokud nastane při provádění databázových operací, které jsou transakčně ošetřeny výjimka, pak jsou veškeré změny provedené v databázi vráceny zpět (tzv. „rollback“). Pokud všechny databázové operace proběhnou v pořádku, pak povolíme zápis provedených změn do databáze (tzv. „commit“).

V rámci návrhu algoritmů, je u operací, které vyžadují, aby byly transakčně ošetřeny toto uvedeno. Zde se jedná většinou případů a záznam nové entity, získání jejího ID a poté záznam její závislosti na jiné entitě (např. vytvoření kategorie, projektu, archívu).

## **4.2 Návrh implementace doplňku GTD**

### **4.2.1 SW prostředky pro implementaci doplňku**

Doplňky (tzv. Add-in) pro sadu MS Office 2007 lze vytvářet pomocí prostředí MS Visual Studio. V úvahu připadají verze Visual Studio 2005, 2008 a nově také 2010.

V případě MS Visual Studio 2005 je možnost využít i bezplatnou edici Express, která však standardně neobsahuje podporu pro vývoj doplňků. Zde je nutno stáhnout rozšíření MS Visual Studio 2005 Tools for Office (VSTO 2005) a nainstalovat.

Novější verze Visual Studio mají toto rozšíření již standardně implementováno, vyjma edice Express.

Pro vývoj doplňku GTD pro MS Outlook 2007 bude použito prostředí MS Visual Studio 2008 Professional Edition.

Jako programovací jazyk pro implementaci doplňku bude použit C#.

Doplňěk bude využívat a pro svou funkci vyžaduje instalaci Frameworku .NET 3.5 SP1 a VSTO Runtime v.9.0.

Doplňěk by se měl přizpůsobovat stylu zobrazení a ovládání MS Outlook 2007 a obecně Windows aplikací tak, aby práce s ním byla intuitivní.

Databáze bude realizována pomocí SŘBD aplikace MS Access 2007, tak aby na koncovou uživatelskou platformu nebyly kladeny nároky spojené s instalací SŘBD.

Pro správnou funkci doplňku tedy bude nutno mít nainstalováno:

- .NET Framework 3.5
- MS Outlook 2007

### **4.2.2 Návrh úložišť a konfiguračních souborů doplňku**

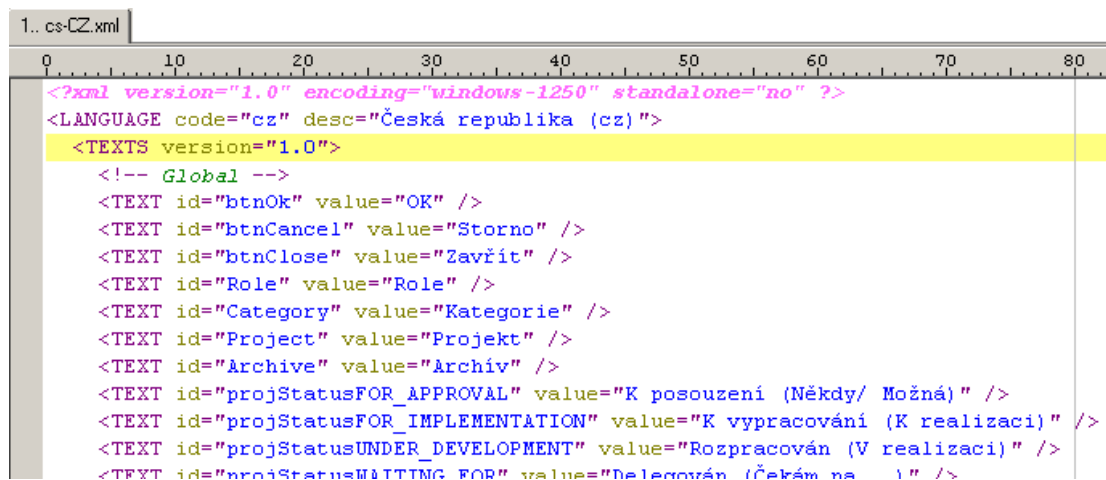
Prostředí a funkčnost aplikace MS Outlook 2007 bude zachována ve stávajícím stavu. Doplňěk bude rozšiřovat pouze možnosti aplikace.

Z důvodu možnosti zachování vlastního nastavení i pro vícero uživatelů, bude databázový a konfigurační soubor doplňku uložen v datovém adresáři aplikace závislém na aktuálně přihlášeném uživateli. Tímto způsobem bude možnost využívat doplňku i na počítači,

který používá více uživatelů. To je samozřejmě možné pouze za předpokladů, že každý uživatel má v systému vytvořený svůj vlastní účet (profil).

Samotná databáze bude představována pouze vlastním \*.acddb souborem. Struktura databáze je pak uvedena v Tabulka 2: Datový slovník (str. 42). Prázdná šablona databáze bude uložena v adresáři „Templates“ v cílovém adresáři doplňku. Po instalaci bude automaticky vytvořena její pracovní kopie v pracovním adresáři aplikace.

Cílový adresář doplňku bude obsahovat adresář pro lokalizaci. Zde se budou ukládat soubory ve formátu XML, které budou obsahovat překlady pro různé jazykové verze aplikace MS Outlook, tak aby veškeré texty v aplikaci MS Outlook, které jsou součástí doplňku, byly shodné s jazykovou interpretací textu v původním MS Outlook. Pojmenování souboru bude totožné s jazykovou verzí MS Outlook (programově zjistitelná pomocí vlastností aplikace Application.CultureName.Name) – jazykový balíček pro češtinu musí mít název „cs-CZ.xml“, aby jej doplněk začal automaticky používat. Pomocí těchto jazykových balíčků bude pro zkušenějšího uživatele aplikace snadné provést lokalizaci doplňku v případě, že jazykový balíček pro daný jazyk není aktuálně k dispozici.



```
<?xml version="1.0" encoding="windows-1250" standalone="no" ?>
<LANGUAGE code="cz" desc="Česká republika (cz)">
  <TEXTS version="1.0">
    <!-- Global -->
    <TEXT id="btnOk" value="OK" />
    <TEXT id="btnCancel" value="Storno" />
    <TEXT id="btnClose" value="Zavřít" />
    <TEXT id="Role" value="Role" />
    <TEXT id="Category" value="Kategorie" />
    <TEXT id="Project" value="Projekt" />
    <TEXT id="Archive" value="Archív" />
    <TEXT id="projStatusFOR_APPROVAL" value="K posouzení (Někdy/ Možná)" />
    <TEXT id="projStatusFOR_IMPLEMENTATION" value="K vypracování (K realizaci)" />
    <TEXT id="projStatusUNDER_DEVELOPMENT" value="Rozpracován (V realizaci)" />
    <TEXT id="projStatusWAITING FOR" value="Delegován (Čekám na ...)" />
```

Obr. 33: Návrh vnitřní struktury lokalizačního balíčku

### **4.2.3 Návrh grafického rozhraní doplňku**

#### **4.2.3.1 GTD v menu aplikace MS Outlook 2007**

Aplikace MS Outlook obsahuje ve svém hlavním okně standardní ovládací prvky, menu a nástrojovou lištu, jež byly použity již v předchozí verzi. Práce s těmito objekty je trochu odlišná od nově zavedeného standardu ovládacích prvků.

Do hlavního menu bude přidána položka „GTD“, která bude obsahovat další položky menu:

- „GTD přehledy“ – zobrazí nebo znovu vyvolá přehledový panel GTD
- „GTD nastavení“ – zobrazí dialogové okno s možnostmi nastavení možnosti pro doplněk GTD

#### **4.2.3.2 Přehledový panel GTD**

Pro zobrazení přehledu o kategoriích, projektech, archívech a samotných objektech aplikace MS Outlook (došlý email, poznámka, připomínka) bude sloužit GTD panel pro reportování, který bude dostupný z úvodní obrazovky aplikace.

Panel bude možno skrýt, přesouvat a měnit jeho velikost tak, aby ostatní dostupné informace, které standardně poskytuje aplikace MS Outlook, byly také k dispozici a dopomáhaly uživateli získat lepší přehled.

Horní část panelu bude obsahovat standardní lištu nástrojů (toolbar) se dvěma comboboxy umožňující výběr role a přehledu (sestav), který chce uživatel zobrazit.

Seznam rolí bude načítán z databáze, přehledy budou mít předem stanové položky k výběru:

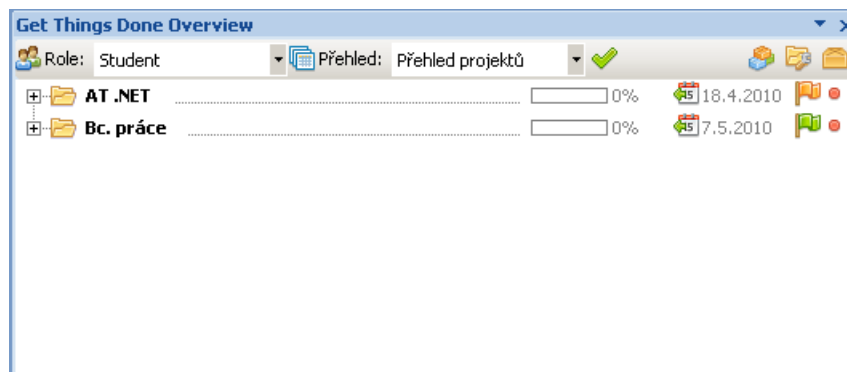
- Přehled kategorií
- Přehled projektů
- Přehled archívů

Dále bude lišta nástrojů obsahovat tlačítko na potvrzení předešlých výběrů. Kliknutím na toto tlačítko se požadovaná sestava zobrazí.

V pravém rohu lišty nástrojů budou umístěna 3 tlačítka:

- správa rolí a kategorií

- správa rolí a projektů
- správu archívů



Obr. 34: Návrh přehledového panelu GTD

#### 4.2.3.3 Nová lišta nástrojů (Ribbon) GTD

Novinka v MS Outlook 2007, která v předešlých verzích nebyla implementována, je zvláštní nástrojová lišta, která může obsahovat další nástrojové lišty, tzv. Ribbon.

Ribbon pro GTD bude zobrazen pouze u vybraných objektů aplikace MS Outlook, a to:

- příchozí e-mail
- připomínka v kalendáři
- úkol

V případě jiných objektů jako odchozí e-mail, poznámka, kontakt atd. nebude lišta GTD zobrazena.

Nástrojová lišta (ribbon) GTD bude obsahovat další ovládací prvky, které budou rozděleny do tří skupin (group):

- Skupina „Role a kategorie“ – bude obsahovat:
  - o tlačítko dynamického menu „Zařadit do rolí“
  - o tlačítko dynamického menu „Zařadit do kategorií“
  - o standardní tlačítko pro správu rolí a kategorií
- Skupina „Projekty“ – bude obsahovat:
  - o tlačítko dynamického menu „Zařadit do projektů“
  - o standardní tlačítko pro správu rolí a projektů
- Skupina „Archivace“ – bude obsahovat



- tlačítko dynamického menu „Zařadit do archívu“
- standardní tlačítko pro správu archívů
- Skupina „Zařazení“ – bude obsahovat:
  - standardní tlačítko „Akceptovat zařazení“
  - standardní tlačítko „Zrušit zařazení“ vedle textového popisu „Kategorie“
  - standardní tlačítko „Zrušit zařazení“ vedle textového popisu „Projekt“
  - standardní tlačítko „Zrušit zařazení“ vedle textového popisu „Archív“

Výsledný vzhled nástrojové lišty GTD může být podobný návrhu na Obr. 18: Návrh lišty nástrojů GTD ze strany 59.

Návrh nástrojové lišty GTD může být proveden pomocí XML konfiguračního souboru nebo pomocí grafického designeru vestavěného v prostředí MS Visual Studio. Co se týče přehlednosti, je výhodnější použití XML definice. Na druhou stranu XML definice potřebuje další podpůrný kód v rámci kódu pro zpracování lišty (ribonu). Navíc, pokud používáme komponentu dynamické menu v této liště (jediná má možnost, jak generovat a měnit svůj obsah dynamicky), je nutné, aby výstup z metody, která zpracovává událost vytvoření menu, byl také XML kód (tj. řetězec, který obsahuje XML kód) definující jednotlivé položky menu. Toto paradoxně zase kód výrazně znepřehledňuje a činí ho v tomto případě zranitelnějším (XML kód uvnitř řetězce nelze spolehlivě ověřit před samotným spuštěním programu), než kdyby se přímo instancovaly objekty třídy „menuButton“ a nastavily jejich parametry. Z tohoto hlediska je výhodnější použití visual designeru. Navíc adresování jednotlivých prvků lišty se pak provádí mnohem snáz, protože vývojové prostředí zná přímo jména objektů (komponent), které byly v liště použité (což při XML konfiguraci přímo nelze).

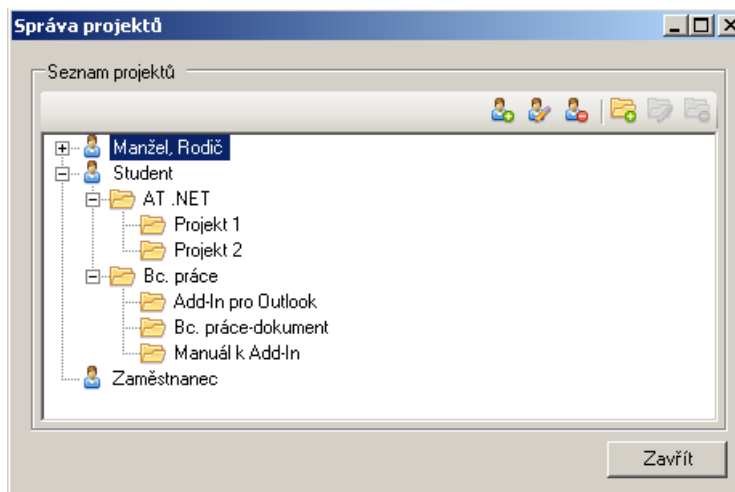
#### **4.2.3.4 Dialogy pro správu kategorií, projektů a archívů**

Pro správu kategorií, projektů a archívů bude vytvořeno nové dialogové okno. Toto okno by mělo obsahovat komponentu pro zobrazení stromové struktury (ať už archívu, projektů nebo kategorií). Nad touto komponentou je nutno vytvořit tlačítka pro správu všech výše zmíněných typů entit. Vespod dialogu je pak nutno umístit tlačítko „Zavřít“ k obvyklému ukončení dialogu a jeho uzavření. Tlačítko „OK“ zde nebude, protože úkony zde provedené budou prováděny pomocí tlačítek pro správu a tyto změny budou okamžitě aplikovány.

Je vhodné zvážit návrh dialogu takovým způsobem, aby bylo možné jej využít pro všechny typy entit (role, kategorie, projekty, archívy) – nabízí se možnost použití třech tlačítek

pro správu rolí (nová, editace a odstranění) a třech tlačítek pro správu entit (nová, editace a odstranění), u kterých by se měnila funkčnost v závislosti na typu entity. V případě, že daná entita je nezávislá na roli (archív), pak tlačítka pro správu rolí je nutno skrýt.

Příklad návrhu dialogu pro správu entit GTD:



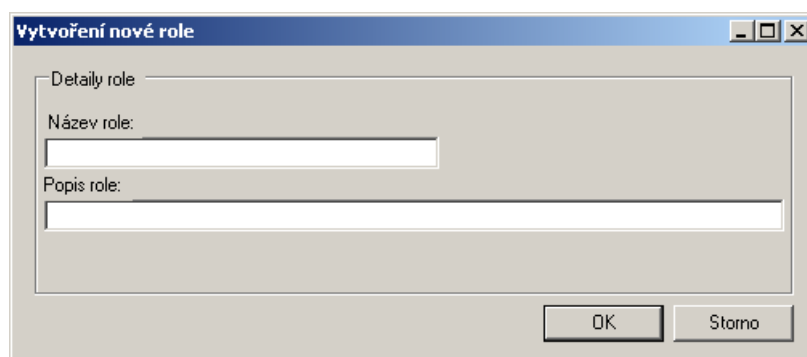
**Obr. 35: Návrh dialogu pro správu entit GTD**

#### 4.2.3.5 Dialogy pro úpravu rolí, kategorií, projektů a archívů

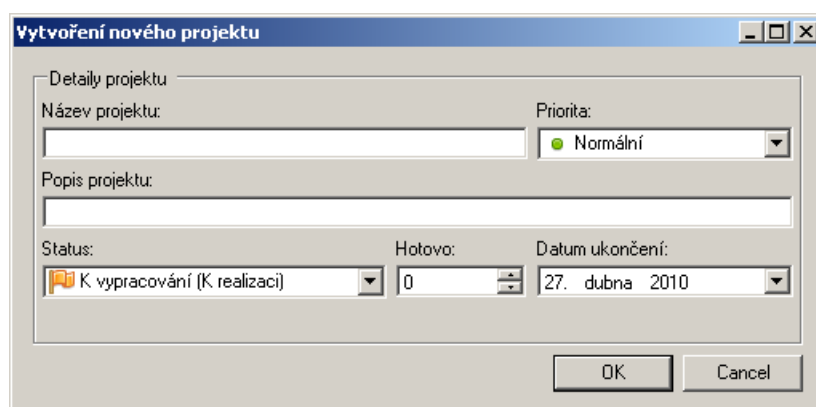
Další dialogové okno je zapotřebí pro úpravy (jako vytvoření nové, změna stávající) samotných entit. Entity typu role, kategorie a archív obsahují shodně pouze dva uživatelem přímo modifikovatelné atributy – jméno a popis (name a description). Z toho vyplývá, že vhodnou implementací tohoto dialogového okna můžeme daný dialog využít opět pro všechny typy entit vyjma projektu. Zároveň pro tyto typy entit můžeme daný dialog využít jak pro vytvoření, tak pro editaci dané entity.

Samotná entita projektu obsahuje trojnásobné množství uživatelem modifikovatelných atributů vzhledem k dříve zmíněným entitám.

Výsledná podoba dialogů může být následující:

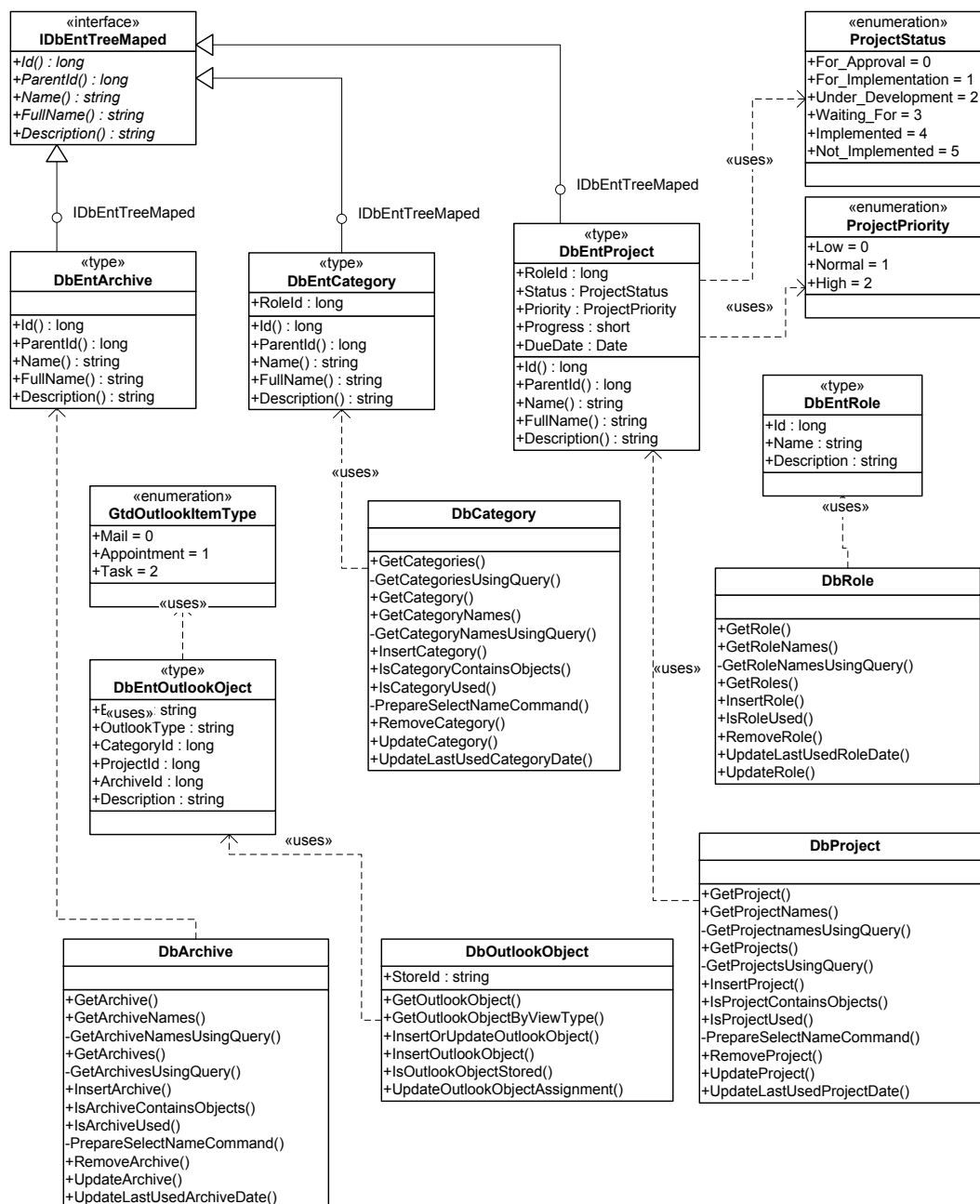


Obr. 36: Návrh dialogu pro vkládání a úpravu podobných entit



Obr. 37: Návrh dialogu na vkládání a úpravu projektů

#### 4.2.4 Návrh implementace rozhraní a tříd



Obr. 38: Třídní diagram pro implementaci Outlook Add-in

Databázové entity budou představovány vlastními třídami. Atributy těchto entit, jsou pak realizovány vlastnostmi těchto tříd. Instance této třídy po vytvoření představuje pak

konkrétní entitu z dané relace a hodnoty atributů jsou uloženy ve vlastnostech dané instance. Pro lepší rozpoznatelnost budou entitní třídy označovány prefixem „DbEnt“.

Relace budou představovány také samostatnými třídami, které budou zajišťovat vlastní operace nad relacemi databáze jako (Select, Insert, Update, Delete). Tyto relační třídy budou využívat instance příslušných entitních tříd jako vstupy a výstupy metod pro operace nad relacemi. Relační třídy budou používat prefix „Db“.

Instanční třídy představující entity kategorie, projekt a archív, mají základní atributy shodné. Toho lze využít pro implementaci rozhraní pro tyto entity. Implementace rozhraní bude např. velkou výhodou při konstrukci stromu pro kategorie, archívy a projekty, kde lze unifikovat pomocí tohoto rozhraní vstupy pro konstrukční metodu tak, že nebude zapotřebí realizovat danou metodu pro každý, z těchto typů entit. Tím dojde k zefektivnění a zpřehlednění kódu.

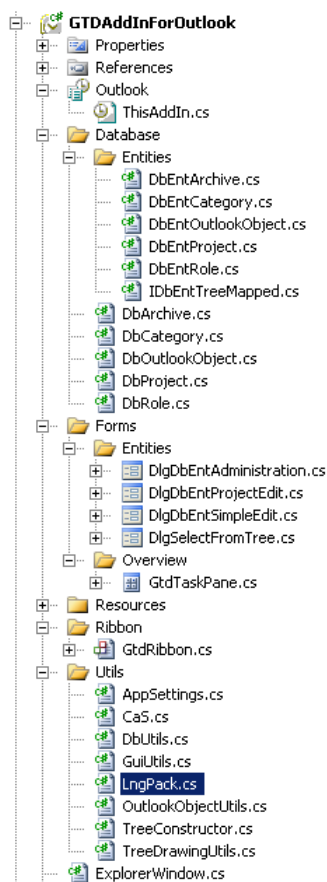
## 4.3 Implementace doplňku GTD

### 4.3.1 Skladba použitých tříd a rozhraní

Rozvržení projektu je navrženo tak, aby jeho jednotlivé části byly udržovány v logických celcích. Toho je docíleno udržováním souvisejících objektů (tříd, dialogů, controls) v různých jmenných prostorech (viz. Obr. 39: Skladba projektu doplňku, kde je oddělení do jmenných prostorů „namespaces“ prezentováno složkami ve kterých jsou uloženy související třídy, atd.)

Rovněž pojmenovávání tříd, dialogů atd. je navrženo tak, aby vypovídalo o funcích, které daná položka poskytuje.

Vytvořený kód je také důsledně komentován pomocí XML komentářů, a tudíž je snáze srozumitelný.



Obr. 39: Skladba projektu doplňku

### 4.3.2 Jmenné prostory

- Jmenný prostor „*GTDAddInForOutlook.Database*“ - slouží k práci s databázovými relacemi. Obsahuje statické třídy, umožňující uskutečňovat operace s danými relacemi z různých míst aplikace bez nutnosti vytváření instancí těchto tříd.
- Jmenný prostor „*GTDAddInForOutlook.Database.Entities*“ – zastupuje dané typy entit v databázi, třídy zde obsažené jsou ve většině případů použity k prezentacím hodnot atributů jednotlivých entit v rámci doplňků
- Jmenný prostor „*GTDAddInForOutlook.Forms.Entites*“ – obsahuje třídy představující dialogy pro zadávání a úpravy hodnot atributů jednotlivých typů entit. Většina dialogů je navržena tak, aby byla univerzální a znovupoužitelná.
- Jmenný prostor „*GTDAddInForOutlook.Forms.Overview*“ – obsahuje controls reprezentující přehledový panel GTD
- Jmenný prostor „*GTDAddInForOutlook.Ribbon*“ – obsahuje kód představující nástrojovou lištu GTD (tzv. ribbon)
- Jmenný prostor „*GTDAddInForOutlook.Utils*“ – obsahuje pomocné třídy. Jde o statické třídy, které usnadňují vykonávání často opakovaných záležitostí (konstrukce stromů, vlastní definice vykreslování, globální konstanty systému, překlady textů do aktuálního jazyka systému apod.)

### 4.3.3 Konfigurační soubor

Vlastní konfigurační soubor doplňku je zastupován třídou „AppSettings.cs“, má XML formát a je uložen v uživatelském profilu aplikace Outlook (stejně jako databáze). Uchovává poslední nastavení pro přehled (výběr role a přehledu) v rámci přehledového panelu GTD a také je zde definován počet položek, které se mají zobrazit v nabídkách menu pro přiřazení rolí, kategorií, projektů a archívů.

```
<?xml version="1.0" encoding="windows-1250" standalone="yes"?>
<SETTINGS>
  <USER_SETTINGS>
    <MENU_COUNT>8</MENU_COUNT>
    <SEL_ROLE>2</SEL_ROLE>
    <SEL_VIEW>1</SEL_VIEW>
  </USER_SETTINGS>
</SETTINGS>
```

Obr. 40: Konfigurační soubor doplňku

### 4.3.4 Získávání EntryID

Každý objekt (email, poznámka, úkol, připomínka v kalendáři) má v MS Outlooku přiřazeno jedinečný identifikátor – EntryID. Entry ID obsahuje identifikační řetězec (není číslem) a je přiřazeno každému objektu MS Outlooku až po jeho uložení, tzn., že v době např. vytváření emailu k odeslání, tento nemá ještě přiřazeno EntryID, to bude přiřazeno až při jeho odeslání (objekt zařazen do složky odeslaná pošta) nebo uložení jako koncept (složka koncepty).

Pro získání objektu v prostředí MS Outlooku ale pouhá informace EntryID nestačí. Metoda GetItemFromID vyžaduje vedle zmíněného EntryID také StoreID identifikátor. Jelikož tento je shodný v rámci plánování klasického použití doplňku v MS Outlooku<sup>7</sup> je v GTD doplňku identifikátor StoreID zjištěn při startu doplňku a poté uložen v globální proměnné.

---

<sup>7</sup> StoreID se však mění se změnou úložiště – další soubor osobních složek, složka MS Exchange serveru, apod.



## 4.4 Řešení problému při implementaci GTD

### 4.4.1 MS Outlook Security Issue

V případě, že potřebujeme přistupovat k emailovým adresám a kontaktům v aplikaci MS Outlook, pak při nesprávné implementaci v doplňku dochází k zobrazení hlášení „Aplikace chce přistupovat ke kontaktům, povolit na jak dlouho?“. Toto je způsobeno právě nesprávně doporučenou implementací, se kterou se běžně můžeme setkat na různých internetových fórech a diskuzích. Zde se běžně pro získání instance aplikace MSOutlook používá přiřazení

`= new Microsoft.Interop.Outlook.Application().`

Takto vytvářená instance by se v doplňku neměla nikdy vyskytovat, jelikož postrádá potřebné bezpečnostní vazby (ověření) a je příčinou onoho bezpečnostního hlášení.

Instanci aplikace MS Outlook bychom vždy měli vytvářet způsobem:

`= Globals.ThisAddIn.Application.`

### 4.4.2 MS Visual Studio a SP1

Pro korektní funkčnost nainstalovaného doplňku je nutno mít nainstalovaný SP1 pro MS Visual Studio 2008. Původní Visual Studio bez SP1 obsahuje chybu (bug) při vytváření Windows instalátoru pro instalaci doplňku pomocí projektu typu „setup“.

Chyba se projevuje v automatické modifikaci hodnoty vlastnosti „Manifest“ v klíči registru HKEY\_CURRENT\_USER \Software\Microsoft\Office\Outlook\Addins\název doplňku. Správná hodnota vlastnosti „Manifest“ musí být nastavena na plnou cestu k souboru ve stylu:

`„[TARGETDIR]GDTAddInForOutlook.vsto|vstolocal“.`

Bohužel při kompilaci setup projektu dojde k automatické modifikaci této hodnoty na:

`„[TARGETDIR]GDTAddInForOutlook.dll.manifest“.`

Tato změna způsobí, že doplněk se ve skutečnosti nenainstaluje a tudíž není v aplikaci MS Outlook 2007 funkční.

## 5 Závěr

Microsoft poskytl prostřednictvím Visual Studio s možností vytvářet doplňky ke stávajícím aplikacím MS Office, efektivní způsob, jak zapojit širší programátorskou veřejnost do rozvoje svých produktů a jak zvýšit přínos svých aplikací koncovým uživatelům.

Vývoj doplňků pro aplikace MS Office je potenciálně velmi mocným nástrojem pro zvýšení produktivity ve společnostech a firmách, které produkty sady MS Office využívají, protože schopný vývojář, obeznámený s problematikou programování doplňků, dokáže poměrně jednoduchou cestou doplnit požadované funkčnosti, bez nutnosti vyvíjet celou aplikaci, jelikož kvalitní základ této aplikace je již dán. Pro vývoj těchto doplňků svědčí i obliba a vysoká míra využívání aplikací sady MS Office.

Obsah předchozích dvou vět zhruba shrnuje i obsah této práce. Původní hybnou silou celého projektu byly požadavky, na rozšíření funkčnosti aplikace MS Outlook o principy nově se šířící metodologie „Getting Things Done“ Davida Allena. Průřez touto metodologií, její hlavní zásady a přínos pro organizaci našeho času shrnuje první část této práce.

V další části, nám práce přináší informace o způsobech a úskalích vývoje doplňků pro aplikace sady MS Office. Mimo seznámení se stručnou historií vývoje aplikace MS Outlook, která byla vybrána jako hostitelská aplikace pro vyvíjený doplněk, jsou zde zmíněny i prostředky pro vývoj doplňků, popsána architektura aplikační vrstvy doplňku, zběžně vysvětlen objektový model aplikace MS Outlook a způsob jakým dokončený doplněk instalovat na klientskou stanici uživatele, včetně uvedení specifických požadavků pro instalaci.

Dále pak práce analyzuje způsob jakým metodologii GTD implementovat do aplikace MS Outlook, navrhuje řešení, tyto řešení rozebírá a blíže specifikuje tak, aby bylo zřejmé jak dané funkcionality implementovat. Rovněž upřesňuje prostředky pro vývoj, navrhuje rozmístění pomocných souborů a podobu uživatelského rozhraní, jakož i skladbu tříd v kódu a nastiňuje další aspekty pro unifikaci kódu aplikace.

Během implementační fáze se pak práce zabývá podrobnostmi v kódu, ukazuje v něm zajímavá a problematická místa. Dokumentuje rovněž způsob vytváření instalačního balíčku, včetně upozornění na některé problémy, které se mohou při vývoji doplňků vyskytnout.

Vytvořený doplněk se snaží přizpůsobit prostředí MS Outlook potřebám metodologie GTD. Každý software, a doplněk je taky pouze a jen software, je nutno využívat a poznávat jeho slabá místa, získávat odezvu od uživatelů, a tyto odezvy pak analyzovat a provádět úpravy

software takovým způsobem, aby byl pro uživatele co nejvíce intuitivní, aby do jejich životů a práce přinášel efektivitu. Takový software pak přináší dvojí zisk, protože spokojený uživatel, je i spokojeným zákazníkem, a jako takový prodlužuje životní cyklus daného software (PLM) a podílí se na vlastně i na jeho vývoji.

Z objektivního hlediska jsou možností doplňků velice široké a je zde mnoho způsobů a příležitostí jak dále doplněk rozšířit, ať už jde o úplně nové funkčnosti, nebo automatizaci stávajících činností. Pro další vývoj GTD doplňku se nabízejí příležitosti plánování projektů v rámci kalendáře MS Outlook, automatizace činností, jako automatické vytvoření vstupu v kalendáři nebo v úkolech z došlé pošty jedním kliknutím apod. Hodně požadavků na vylepšení pak vzejde i z praxe při každodenním používání doplňku v aplikaci MS Outlook. Těmito směry, já osobně, vidím další vývoj aplikace.

## Seznam použité literatury

- [1] ALLEN, David. Mít vše hotovo, 1. vydání, 2008.  
Jan Melvil Publishing, ISBN 978-80-903912-8-4, str. 7, 43.
- [2] Mojave Media Group., *Microsoft Outlook: A Brief History [online]*. 14. Srpen 2008.  
<<http://www.brighthub.com/office/collaboration/articles/1491.aspx>>
- [3] CABRAL E.E. – FOX, *Building Office Bussiness Applications Using Visual Studio Tools for the Office System (3.0)*. Leden 2008. < [http://msdn.microsoft.com/en-us/library/bb981164\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/bb981164(v=office.12).aspx)>
- [4] *Architecture of Application-Level Add-Ins, 2010*.  
<<http://msdn.microsoft.com/en-us/library/bb386298.aspx>>

# Seznam příloh

Příloha I      GTD Add-In for Outlook - Uživatelský manuál